ARAB ACADEMY FOR SCIENCE, TECHNOLOGY AND MARITIME TRANSPORT

College of Engineering and Technology

Computer Engineering Department

Natural-Inspired Drone Swarm for Efficient Multi-View Monitoring and Object Detection

By

OSAMA HESHAM ELSAYED ABDEL RAZEK

B.Sc. Computer Engineering AASTMT, 2019

A thesis submitted to AASTMT in partial

Fulfillment of the requirements for the award of the degree of

MASTER'S OF SCIENCE

in

COMPUTER ENGINEERING

Supervised by

Prof. Dr. Sherine Mostafa Youssef **Prof. Dr. Ossama Ismail**

Computer Engineering Department

College of Engineering and Technology

Arab Academy for Science, Technology and Maritime Transport

Alexandria, Egypt

2023


بسم الله الرحمن الرحيم

DECLARATION

I certify that all the material in this thesis that is not my own work has been identified, and that no material is included for which a degree has previously been conferred on me.

The contents of this thesis reflect my own personal views and are not necessarily endorsed by the University.

Name Osama Hesham El Sayed Abd El Razek

Signature 

Date 20-July-2023

ACKNOWLEDGMENT

Many alhamdulillah for granting me his blessings and facilitating this thesis for me. I am deeply thankful to Allah for giving me the will and patience to complete this work.

I would like to express my sincere gratitude to my advisor Prof. Dr. Sherine Youssef for the continuous support of my thesis study and research, for her motivation, enthusiasm, valuable advice, and guidance throughout the thesis.

My heartful thanks and deepest gratitude also goes to my second advisor Prof. Dr. Ossama Ismail for his unlimited support, help, patience, and guidance. His insightful suggestions, comments and encouragement helped me complete this thesis.

I am also sincerely thankful to my kind supportive family for always supporting and helping and their unconditional love and patience, I would not have made it without them.

Many thanks to everyone who supported and helped me in completing this work whether with resources, advice, or time.

DEDICATION

Dedicated to my loving and supporting wife and family.

ABSTRACT

In recent years, the use of unmanned aerial vehicles (UAVs) or drones has significantly increased in various fields, including surveillance applications. However, the use of multiple drones for swarm missions can be costly and complicated, making it difficult to manage and monitor the data generated. In this regard, the Natural-Inspired Drone Swarm for Efficient Multi-View Monitoring and Object Detection, proposed a cost-effective alternative using commercially available Tello drones for simultaneous control of multiple drones and real-time video stitching. The study aims to decrease data transfer, storage, management, and monitoring by introducing efficient multi-view panoramic imaging and extra compression of surveillance swarm drones' cameras' footage through stitching.

To achieve this goal, the study used camera path estimation and homography refinement method to enable simultaneous control of multiple drones and real-time video stitching. Additionally, the researchers proposed a unified framework for joint video stitching and stabilization, which involves creating an efficient virtual 2D camera path, space-temporal optimization, grid-based tracking, and mesh-based motion models, the study conducted three experiments to evaluate the effectiveness of the proposed approach. Based on the results, the stability score was consistently high across all three experiments, indicating relatively smooth and stable stitched videos. However, the stitching score varied across the three trials, with the first trial having a relatively low score, indicating good alignment and high-quality stitching. In contrast, the second and third trials had higher stitching scores, indicating lower quality stitching with lower alignment.

In summary, the proposed approach of using commercially available Tello drones for swarm missions with real-time video stitching and stabilization is an effective and cost-efficient method for surveillance applications. The study's results showed that the approach achieved high stability scores while maintaining a acceptable level of stitching quality, making it a promising solution for managing and monitoring large amounts of data generated by multiple drones.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS/ABBREVIATIONS

CCD	Charged-Coupled Device
CCTV	Closed-Circuit Television
CNN	convolutional neural network
COCO	Common Objects in Context
FAST	Features from Accelerated Segment Test
FOV	Field Of View
GPS	Global Positioning System
ICE	Image Composite Editor
KLT	Kanade–Lucas–Tomasi feature tracker
LPVW	Line-Preserving Video Warp
LQR	Linear–Quadratic Regulator
MAP	Mean Average Precision
MAV	Micro Ariel Vehicles
MSE	mean square error
PSNR	peak signal-to-noise ratio
RADAR	Radio Detection and Ranging
RANSAC	Random Sample Consensus
RGB	Red, Green, and Blue
RPI	Raspberry Pi
SIFT	Scale-Invariant Feature Transform
SURF	Speeded-Up Robust Features
SDK	Software Development Kit
UAV	Unmanned Ariel Vehicles
YOLO	You Only Look Once

CHAPTER ONE
INTRODUCTION

1 INTRODUCTION

Unmanned aerial vehicles (UAVs), commonly referred to as drones, are garnering greater attention in both academia and industry, (see figure 1-1, which shows different types of UAVs). This is due to advances in sensing and computing capabilities, as well as reductions in form factors and costs. Moreover, the development of onboard intelligence and autonomous capabilities has expanded the range of applications for UAV systems. In recent years, the combination of UAV and swarm intelligence technologies has enabled swarms of small-scale and low-cost UAVs to execute complex combat tasks **[1]**. Moreover, the problem statement and the main objectives of the thesis will be provided. A brief description of the thesis layout is given at the end of this chapter.

Figure 1-1 Different UAV types

1.1 What are Micro Drones?

Micro aerial vehicles (MAVs) are small and lightweight unmanned aerial vehicles that have greatly contributed to the development of robotics and unmanned systems, (see figure 1-2, which shows different models of MAVs). They are widely used for various applications such as surveillance, search and rescue, and mapping. One of the significant advantages of MAVs is their portability, which allows for easy transportation and deployment in different locations. Their small size also makes them more agile and able to maneuver through narrow spaces while reducing the risk of causing damage to their surroundings **[2]**.

However, the smaller size of MAVs also limits their capabilities. They have shorter flight times, less on-board sensing and computing power, and lower payloads compared to larger drones. This makes it challenging for MAVs to carry out complex tasks on their own **[1] [2]**.

Figure 1-2 Models of Micro drones

1.2 Swarm of MAVs

To address these limitations, researchers have developed the concept of aerial swarms. A swarm of MAVs can work together to perform complex tasks by collaborating and overcoming the limitations of individual robots. This approach allows the swarm to achieve tasks that would be difficult or impossible for a single MAV to complete alone, such as mapping large areas, inspecting infrastructure, and even delivering goods, (see figure 1-3, which shows a swarm of MAVs). Overall, the development of MAVs and aerial swarm technology has opened up new possibilities for unmanned aerial systems in various fields **[1] [2]**.

Figure 1-3 Swarm of MAVs

In a swarm configuration, each drone can be assigned specific data collection and processing tasks and equipped with enough computing power to execute these tasks in real-time. However,

for more computationally intensive tasks, the central processing may occur on a more powerful server or base station, or even in the cloud. This allows for more efficient data processing and analysis and can enable more complex tasks to be carried out by the swarm as a whole **[3]**, (see figure 1-4, which shows a swarm of MAVs communicating through the cloud).

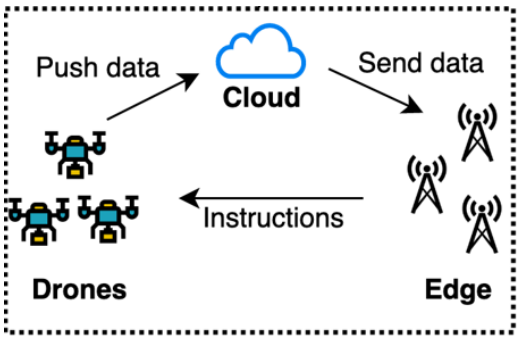

Figure 1-4 The communication and retrieving of data from UAVs

Drone swarms offer several benefits, including increased efficiency in covering larger areas more quickly and efficiently than individual drones **[2]**. They can reduce costs and provide real-time data collection and analysis, enabling quicker decision-making and more effective responses to emergencies. Drones can perform tasks that would be dangerous for humans and be easily reconfigured and adapted to different tasks and missions, making them highly versatile and adaptable **[3]**. Using drones for data collection can also reduce the environmental impact of traditional methods, such as using manned aircraft or ground vehicles. The benefits of using drone swarms are numerous and are leading to exciting new applications across a range of industries **[4]**.

1.3 Real-time stitching

Real-time video stitching is a process that involves seamlessly combining video feeds from multiple cameras or drones in real-time to create a single panoramic or wide-angle view. This technology is often used in situations where a comprehensive view of a large area is required, such as in surveillance, security, or disaster response scenarios **[5]**.

The real-time video stitching process involves several steps, including capturing video feeds from multiple cameras or drones, calibrating the cameras to ensure they are properly aligned, and stitching the video feeds together into a single panoramic view. This requires sophisticated

software algorithms that can adjust for differences in camera angles, lighting, and other factors to create a seamless and accurate view of the area being monitored **[6]**, (see figure 1-5, which shows the video stitching example).

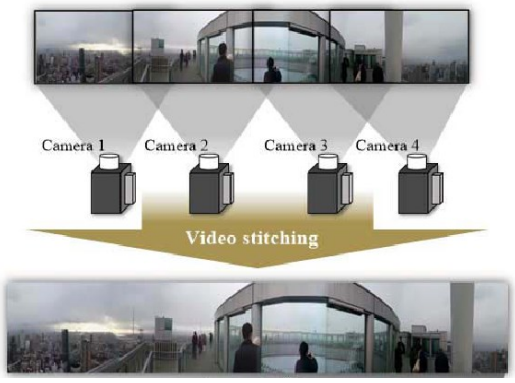

Figure 1-5 Video stitching input and output

Real-time video stitching can provide several benefits, including improved situational awareness, faster decision-making, and reduced costs. By providing a comprehensive and seamless view of the monitored area, operators can quickly identify potential issues or threats, enabling them to respond quickly and effectively. Additionally, real-time video stitching can reduce the need for multiple cameras or operators, further reducing costs **[5]**.

Overall, real-time video stitching is a powerful technology that can provide a comprehensive view of large areas in real-time, enabling faster decision-making and improving situational awareness for various applications **[6]**.

1.4 Major Challenges In Aerial Object Detection

Aerial object detection has some interesting characteristics which makes it a much more difficult task than regular view object detection. We highlight some of the major challenges to aerial object detection **[7]**, see figure 1-6.

- Low spatial resolution: Aerial images captured by drones from high altitudes result in small objects being captured with low spatial resolution. This makes it challenging for object detection algorithms to detect and classify each object separately with distinctive boundaries. Additionally, crowded objects in the image further complicate the task **[7]**.

- Multitude of object sizes: Drones capture a large field of view resulting in objects of varying spatial dimensions being captured in a single image. Objects like trucks and buses are easier to detect due to their larger size, whereas smaller objects like bicycles and awning tricycles are harder to detect as the resolution keeps decreasing as we go deeper into the neural network. Hence, handling different varieties of objects together is a challenge for deep learning-based models **[7]**.

- Occlusion and variation in surrounding illumination: Occlusion and shadowing of objects by large buildings and trees make detection more difficult in aerial images. Due to the small size of the objects, even partial occlusion can make the detection of such objects almost impossible. Variation in surrounding illumination, with some objects lying in illuminated areas and others in dark regions, also adds to the difficulty **[7]**.

- Limited computational resources: The limited computing resources and battery power of UAVs make it difficult to carry out deep learning inference in the UAVs. Therefore, some or all of the computations are offloaded to other devices. Achieving high inference accuracy and lower delay still remains a challenge. There are two methods of offloading: airborne offloading, where the UAV offloads its computing tasks to a nearby UAV with available computing resources, and ground offloading, which allows tasks to be offloaded to an edge cloud server. Some proposed algorithms for efficient offloading deploy lower layers of the CNN on the UAV, while deeper layers are deployed on the MEC server. However, there is still work to be done to achieve high inference accuracy and lower delay **[7]**.

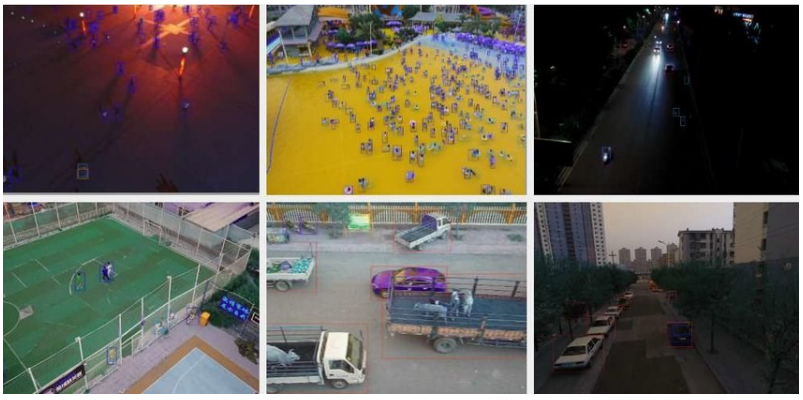

Figure 1-6 Challenges in aerial object detection

1.5 Acquiring & Processing Drone Images

To capture terrain and landscapes comprehensively, acquiring aerial images typically involves two main steps:

- Photogrammetry: During a UAV flight, multiple images are captured at regular intervals, with a significant amount of overlap between the images. This overlap is crucial for measurements between objects present in the images to be made. This process is called photogrammetry, and relevant metadata is automatically inserted by a microcomputer onboard the UAV to enable the images to be used for data analysis and mapmaking.

- Image Stitching: Once the data acquisition is complete, the next step is to combine individual aerial images into a useful map. A specialized form of photogrammetry called Structure-from-Motion (SfM) is commonly used to quickly stitch images together. SfM software stitches images of the same scene from different angles by comparing, matching, and measuring angles between objects within each image. During this step, the images may also be geo-referenced to attach location information to each image.

After image stitching, the generated map can be used for various types of analysis for the applications mentioned above, such as identifying changes in land cover, monitoring urban development, or creating 3D models. Additionally, advanced techniques such as multi-view stereo can be applied to the images to create detailed elevation models, which can be used for further analysis and mapping applications.

1.6 Motivation of Automating The Analysis of Aerial Imagery

Automating the analysis of aerial imagery can enable faster and more efficient processing of large amounts of data, making it possible to identify patterns and trends that may not be apparent through manual analysis. For instance, with the help of machine learning algorithms, it is possible to classify urban land use based on features extracted from aerial images, such as the size and shape of buildings, the presence of vegetation, and the density of road networks. Similarly, machine learning can be used to detect changes in the environment, such as deforestation, urbanization, and land degradation, by comparing images taken at different points in time. Additionally, machine learning algorithms can be used for crowd counting and tracking,

enabling more accurate and timely predictions of crowd behavior during large public gatherings. Overall, the integration of artificial intelligence and drones has the potential to revolutionize many areas, including agriculture, transportation, public safety, and environmental monitoring, among others.

These are certainly significant challenges that need to be addressed when automating the analysis of drone imagery. In addition to the ones mentioned, there are other issues to consider such as lighting conditions, weather conditions, and variability in the data due to different sensors and camera systems. These factors can affect the quality of the images and the accuracy of the algorithms used to process them.

As the field of aerial imagery analysis continues to evolve, it is likely that new approaches and techniques will emerge to overcome these challenges and enable more accurate and efficient analysis of drone imagery.

1.7 Problem Statement

Increasingly, drone swarms and real-time video stitching technologies are being employed to provide a comprehensive view of large areas, offering several advantages, such as heightened situational awareness, improved operational efficiency, cost savings, and enhanced safety outcomes **[5]**.

Drone swarms can rapidly and efficiently cover expansive areas, reducing the need for human intervention, and often proving more cost-effective than traditional monitoring methods. They find valuable applications in monitoring hazardous areas like search and rescue operations or disaster response, thereby minimizing risks to human operators **[5]**.

Real-time video stitching seamlessly merges video feeds to offer a unified view of the monitored area. This facilitates quicker decision-making, reduces the requirement for multiple cameras or operators, and enables more informed decisions, ultimately leading to improved safety outcomes **[8]**.

In conclusion, the adoption of drone swarms and real-time video stitching technologies holds the potential to elevate situational awareness, enhance efficiency, cut operational costs, and elevate safety across a wide range of applications.

1.8 Motivations and Objectives

The use of Tello drones for drone swarms and real-time video stitching can offer several benefits and opportunities for various applications. One of the main motivations for using these technologies is to improve situational awareness. By deploying a swarm of Tello drones and using real-time video stitching technology, operators can obtain a comprehensive and seamless view of the monitored area. This can help to identify potential issues or threats, enabling quick and effective responses **[8]**.

Another motivation is to enhance efficiency. Tello drones can cover large areas quickly and efficiently, reducing the need for human intervention. By using real-time video stitching, operators can analyze footage from multiple drones simultaneously, enabling faster decision-making. This can lead to better outcomes and increased productivity **[9]**.

Cost savings is another objective of using Tello drones and real-time video stitching. The use of drones can be more cost-effective than traditional methods of monitoring large areas, such as manned aircraft or ground-based patrols **[9]**. Additionally, real-time video stitching can reduce the need for multiple cameras or operators, further reducing costs **[8]**.

In summary, the use of Tello drones for environmental monitoring, in combination with real-time video stitching, provides specific opportunities for research and development in the context of environmental data collection. These applications not only enhance situational awareness, efficiency, and cost savings but also open doors for advancing the effectiveness of these technologies in the field of environmental science.

1.9 Challenges

Developing Tello drone swarms and real-time video stitching from drones poses several challenges. Communication and coordination between individual drones can be difficult due to their limited on-board processing capabilities. Effective communication and coordination protocols are crucial to ensure successful swarm operations. Real-time video stitching from multiple drone cameras is challenging due to differences in camera positions, angles, and lighting conditions. Developing accurate algorithms for stitching together video feeds from multiple drones in real-time requires sophisticated image processing techniques. Additionally, transmitting and storing the large volume of data generated by multiple drones during video

capture is a challenge that requires the development of effective data transmission and storage solutions. Addressing these challenges requires the development of sophisticated hardware, software, and communication solutions that can operate within the processing constraints of Tello drones.

1.10 Thesis Contributions

The proposed model for joint stitching and stabilizing live video feeds from multiple quadcopters makes a significant contribution in the field of drone-based video monitoring. By estimating the inter and intra motion between live feeds, the model addresses the challenges of achieving spatial alignment and temporal smoothness in the stitched video.

The optimization approach used in the model ensures the best fit between the different video feeds and provides a high-quality output that is both stable and smooth. The method of dividing video frames into smaller cells makes it easier to use the bundled-path methodology, and the handling of scenes with parallax is a valuable addition to the model's capabilities.

The model's exceptional object detection performance, as evidenced by high average precision measures across multiple recall levels, is a significant contribution. Additionally, the model's consistent and robust performance in diverse experimental settings, with relatively high stability and stitching scores, indicates its effectiveness and potential for practical use in real-world applications.

Overall, the proposed model makes a valuable contribution to the development of drone-based video monitoring and has the potential to improve situational awareness and enhance decision-making in various settings, including surveillance, security, disaster response, and more.

1.11 Thesis Outline

The thesis is organized as follows:

Chapter 2: lists the previous work done in video stitching and swarm of micro aerial vehicles.

Chapter 3: provides the needed background for the reader to understand different terminologies used in the thesis.

Chapter 4: gives a detailed explanation for the proposed natural-inspired drone swarm processing FOV for efficient multi-view monitoring and object detection.

Chapter 5: describes all the experiments conducted to validate the proposed model. It also illustrates the performance indicators used in evaluating the system's performance and the experimental results and provides a comparison with previous studies.

Chapter 6: presents a summary, conclusion of the thesis and future work directions.

CHAPTER TWO
LITERATURE REVIEW

2 LITERATURE REVIEW

The use of drone swarms for real-time video stitching has gained significant attention in recent years, with various methods proposed to address the challenges of camera motion, lighting conditions, and scene complexity. These methods include feature-based and optical flow-based techniques, as well as deep learning-based approaches. The proposed methods have been evaluated on various datasets, including indoor and outdoor scenes, and have achieved high-quality and robust results. However, some limitations, such as the requirement for a static scene and high computational resources, have been reported. Overall, the advancements in this field have shown great potential for the use of drone swarms in applications such as surveillance, search and rescue, and disaster response, (see figure 2-1, which demonstrate the uses of UAVs).

2.1 Overview Of Aerial Swarm Applications

The use of aerial swarm applications has expanded significantly in recent years, with the development of advanced technologies in areas such as drone swarms, machine learning, and computer vision. These applications range from aerial surveillance and reconnaissance to disaster response and environmental monitoring. Drone swarms can be used for tasks such as mapping, search and rescue, and infrastructure inspection, with the potential to provide real-time data and situational awareness in challenging environments. The use of drone swarms in applications such as precision agriculture and wildlife conservation has also been explored, with the potential to improve efficiency and reduce costs. Overall, the advancements in aerial swarm applications have opened up new possibilities for various industries, with the potential to address critical challenges and improve decision-making processes.

Figure 2-1 Demonstrate the well-knows uses of UAVs

- **Security and Surveillance**

The adoption of drones in industrial and commercial security applications has increased significantly with recent advancements in drone and AI technologies. Drones can now be equipped with advanced onboard monitoring sensors such as RGB and thermal cameras, making efficient and effective aerial surveillance possible. The advantages of surveillance drones over traditional security methods, such as fixed CCTV cameras and human patrols, include the ability to cover larger areas, reduce costs associated with human patrols, and minimize risks to human security personnel. The use of drone swarms in security applications further enhances these advantages by enabling simultaneous monitoring of multiple areas, resulting in significant time savings, and increased operational efficiency, (see figure 2-1, which shows a drone equipped with cameras scanning areas and communicate with a command center). As drone and AI technologies continue to advance, it is expected that the deployment of drones for industrial and commercial security will become increasingly widespread, revolutionizing the way security is managed and maintained around the world [10] [11].

The use of UAV fleets can optimize area coverage, adapt to changes in the environment or mission objectives, and collaborate to accomplish surveillance tasks. With advanced onboard intelligence and communication capabilities, the UAV fleet can adjust its behavior and strategies in real-time, making it a powerful tool for surveillance operations [10]. Furthermore, the use of UAV swarms can enhance their capabilities by enabling more efficient and effective surveillance over larger areas in a shorter amount of time [11]. Detecting and tracking objects can be achieved by employing individual sensor technologies such as radio frequency (RF) detection and spoofing, RADAR, optical sensors including RGB and thermal cameras, and acoustic sensors [12] [13]. The combination of these technologies can result in increased accuracy of detection and tracking. Although swarm-based anti-UAV systems have not yet been commercially developed, research efforts are being conducted in this area. The use of UAV swarms can also be advantageous in surveillance applications where large areas need to be covered and searched in shorter periods compared to single UAVs [12] [13].

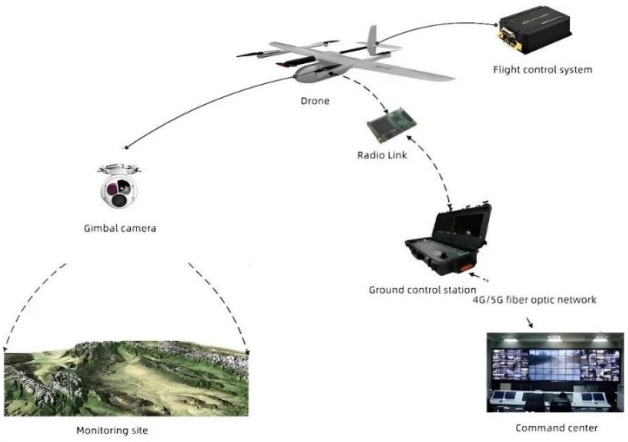

Figure 2-2 UAV equipped with camera for surveillance tasks

- **Collaborative Transportation**

Payload transportation using a swarm of UAVs is a promising area of research due to its potential to overcome the payload limitations of single UAVs. However, safety and regulatory constraints make it difficult to deploy large and heavy UAVs for transporting heavier payloads. Therefore, researchers have demonstrated the use of a group of small UAVs working together for transporting larger payloads. Decentralized control laws were used for each quadrotor, where each quadrotor knows its fixed relative position and orientation with respect to the body and payload goal [14]. The required state estimation of quadrotor positions and velocity was done in a centralized way using an overhead motion capture system. In addition, methods to estimate payload deformation and stabilize it in 3D using a centralized LQR controller were presented for transporting a certain class of flexible structures that were still rigidly attached to the UAVs. The system demonstrated the feasibility of using onboard sensors for outdoor multi-UAV payload transportation, (see figure 2-2, that shows multi-UAVs carrying payload). The use of a decentralized approach where each quadrotor estimates its own pose using visual-inertial odometry and communicates with its neighbors to maintain a desired formation is also promising for real-world applications in various outdoor environments [15].

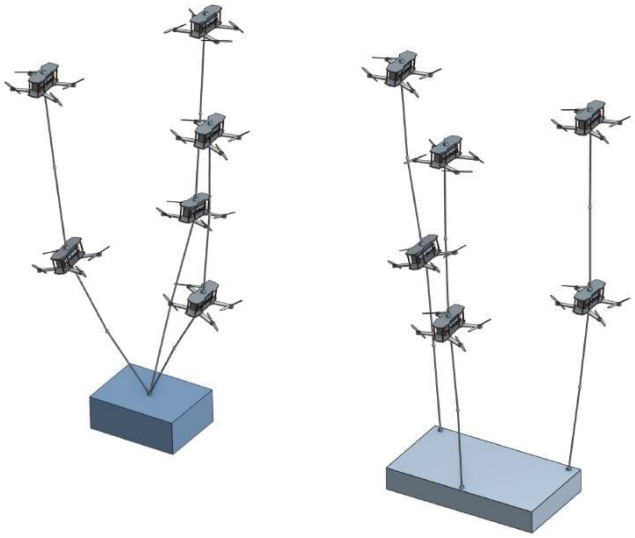

Figure 2-3 UAV carrying a delivery package

- **Environmental Monitoring**

A swarm of UAVs equipped with various sensors, including optical and thermal cameras and GPS receivers, was deployed to monitor and map flood-prone areas. The UAVs autonomously flew over the flood zone and captured high-quality images and videos, which were transmitted to a ground station in real-time. The system processed and analyzed the data to create accurate flood maps and track the movement of the floodwater, facilitating timely and efficient responses from disaster management authorities. The use of a UAV swarm enabled the coverage of a larger area in a shorter period, which is crucial in the case of rapidly changing flood events [16], (see figure 2-3, which shows UAV detecting gasses in certain area).

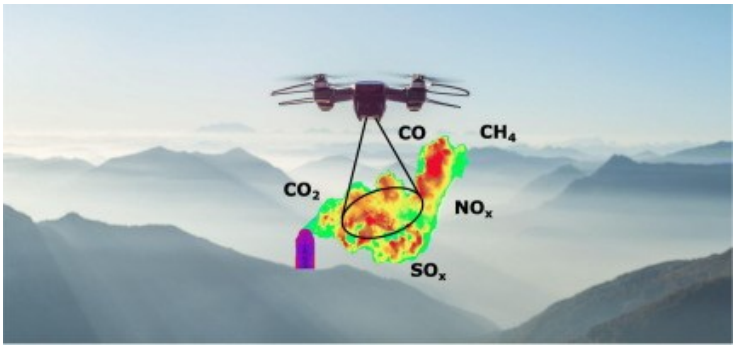

Figure 2-4 shows UAV monitoring gases

2.2 Real-time Stitching Techniques

Various software packages such as Adobe Photoshop, AutoStitch, PTGui, and Image Composite Editor (ICE) are widely used to stitch multiple overlapping images together to generate a wide-angle view. These software packages utilize a combination of feature detection, matching, and alignment algorithms to automatically align and blend multiple images into a seamless panorama [5].

There are also 360-degree polydioptric cameras available on the market, such as Nokia OZO', GoPro Odyssey', Facebook Surround 360', and Samsung Gear360', that capture a complete 360-degree view of the scene for virtual reality (VR) and immersive media applications, (see figure 2-4, which shows the different models of cameras available in the market). These cameras capture images that are stitched together using specialized software to create a seamless 360-degree panorama [5].

Figure 2-5 shows different models of commercial cameras

In addition, Google Street View is an application that uses CMOS cameras to capture video at lower sampling rates, resulting in less continuous videos than regular ones. This application utilizes specially designed head cameras with fixed relative geometries that are synchronized for capturing. Video stitching is a crucial aspect of this application [17].

Image and video stitching are techniques used to combine multiple images or videos into a single panoramic or wide-angle view. The process involves matching overlapping portions of the images or videos and then blending them together seamlessly to create a cohesive and

continuous image or video. In image stitching, feature detection algorithms are used to identify common points between images, and a transformation matrix is applied to adjust the perspective and align the images. In video stitching, the same process is applied to each frame of the video to create a seamless panoramic video. The resulting stitched image or video, (see figure 2-6, which shows the common process for stitching images and videos).

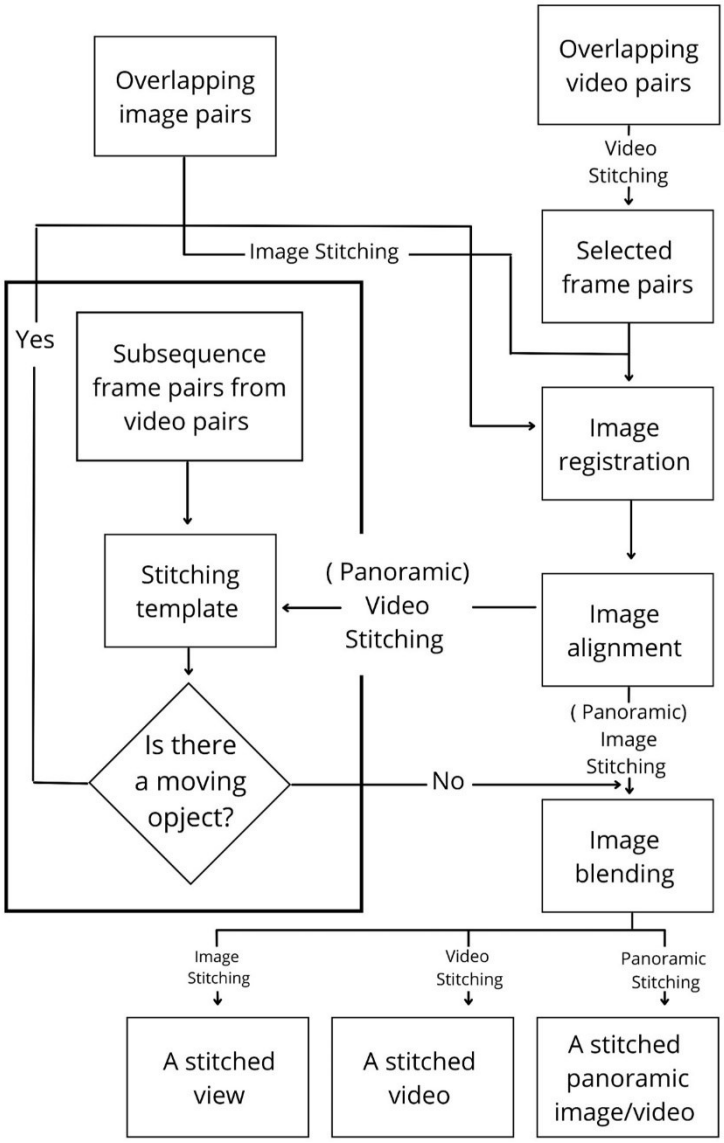

Figure 2-6 The process of stitching videos and images

The stitching process differs on the cameras motion, there are two types of camera positioning:

2.3 Static Cameras

The process of real-time video stitching has gained significant attention in recent years, especially with the emergence of 360-degree cameras and virtual reality technologies [5]. One major challenge in video stitching is dealing with moving objects and their effects on the stitching process. Several approaches have been proposed to address this challenge [18] [19]. One such approach is to embed the detected moving objects into the final stitched images using a standard stitching algorithm and then perform an object detection phase to ensure the accuracy of the embedding phase. Another approach involves using spatially neighboring videos captured by a horizontally swiveling camera from a stationary location to provide reliable information for accurate alignment between frames [20].

In addition to these approaches, a surveillance application video stitching model has been introduced that utilizes a coarse-to-fine process. This involves creating different layers from the chosen frames obtained from the input videos and then stitching the background layers using a standard stitching pipeline algorithm, as long as there are no moving objects, and all the objects are static across the overlapping regions of the stitched frames. Multiple layers containing variant objects are generated from the clustering of the matched feature pairs, where each generated layer has a collection of matched feature pairs consistent with the identical homography, while the videos are pre-aligned [18].

To address issues such as missing data, artifacts, or ghosting caused by moving objects, researchers have proposed adjusting the best fit seam by analyzing the gradient variations in the overlapping regions. Another approach involves employing a spatial-temporal mesh optimization framework to improve the geometric alignment between frames by formulating an objective function that captures the matching costs in both domains. This approach uses initially aligned frames of the input videos based on their estimated spatial and temporal transformations and assigns higher weights to salient regions such as spatial and temporal edges to obtain the optimal seam and minimize the salient effect [21].

In conclusion, the field of real-time video stitching for static cameras has seen significant progress in recent years with the development of several approaches to address the challenges posed by moving objects. These approaches range from coarse-to-fine processes to mesh optimization frameworks and have resulted in improved performance metrics such as accuracy

and efficiency. However, limitations such as the need for pre-alignment and the sensitivity to salient regions and edges continue to be areas of ongoing research [19].

2.4 Moving Cameras

The use of unmanned aerial vehicles (UAVs) and smartphones for capturing videos has led to an increase in the use of real-time video stitching techniques. However, moving cameras often result in shaky and distorted videos, making video stitching and stabilization a significant challenge [21]. To address these issues, many studies have proposed combining image stitching and stabilization techniques to achieve better results.

One proposed solution involves a two-fold transformation approach that involves an inter-transformation between the two cameras to establish spatial alignment, followed by an intra-transformation conducted within each video to maintain temporal smoothness. A smooth virtual camera path is then synthesized to achieve the required video alignment for stitching, followed by a bundled-paths method for video stabilization. The proposed model uses an objective function that combines a stabilization term and a stitching term, which is optimized using an iterative optimization scheme to obtain the final output [19].

Another approach involves combining a dense 3D reconstruction and camera pose estimation technique to stitch input videos from hand-held cameras. This approach requires recovering 3D camera motions and sparse scene points using overlapping scenes to reconstruct 3D scenes, followed by constructing a smooth virtual camera path that remains in the middle of all the original paths. The Line-Preserving Video Warp (LPVW) method is then used, which utilizes a mesh-based warping optimization to synthesize the stitched video while simultaneously stabilizing it [22].

Video stitching algorithms typically involve several steps, including constructing a stitching template by stitching selected frames of original videos with image stitching algorithms, which is then used to stitch subsequent frames to generate a single wide-angle video. To address potential blurring and ghosting in the stitched video, foreground detection is often employed to identify and isolate moving objects in the scene [5].

Overall, real-time video stitching techniques have become essential in various fields, including autonomous vehicles, virtual reality, and surveillance. With the advancement in

computer vision and image processing techniques, there is a growing interest in exploring new methods for multi-view video stitching to achieve better results [18].

Multi-view video stitching is the process of combining multiple video clips captured from different angles to create a panoramic video with a wider field of view. The challenges in this process are different from those in multi-view image stitching because of the added temporal properties of video, such as video synchronization and video stabilization. To address these challenges, computer vision solutions are being developed to robustly align frames from the temporal sequence [8].

In addition to spatial stitching problems, video stitching also involves video processing problems, particularly when the input videos are captured by mobile devices that may introduce jitter. Effective handling of camera motions is crucial in video stitching, and video stabilization techniques can eliminate shaky and jittery movements [8].

In contrast to humans, robotic systems are suitable for hazardous environments where human safety is a concern, such as in nuclear or chemical industries. These systems do not require the same environmental conditions as humans, such as lighting, air conditioning, or noise protection. Moreover, robots have advanced sensors and actuators that can surpass human capabilities in certain aspects. Mesh networking and redundant data storage make robotic systems very fault tolerant and hard to bring down, while swarms of robots are highly scalable and self-organizing, making them suitable for large-scale exploration and monitoring [23].

The literature on video stitching highlights several major challenges, including video stabilization, video synchronization, efficient large-size multi-view video alignment and panoramic video stitching, color correction, and blurred frame detection and repair.

Table 2-1 Summarizes part of the work done in this area

Method	Description	Results	Limitations	Performance Metrics
Real-Time Video Stitching Using Camera Path Estimation and Homography Refinement [24]	The Paper Proposes a Real-Time Video Stitching Method Using CP Estimation and Homography Refinement to Reduce Processing Time and Remove Misalignments in Handheld Camera Videos. The Proposed Method Employs a Grid and Optical Flow for	The Proposed Method Approximately Shortens the Processing Time by A Factor Of 50. This Confirms That the Proposed Method Is Suitable for Real-Time Systems. The Homography Refinement Results in An Average Stitching	The Proposed Method Has Limitations When Dealing with Dramatic Camera Movements and Multiple Planes in The Scene. It May Require Additional Components Such as Video Synchronization and Stabilization to Improve Its Performance.	Processing Time and Stitching Quality

	CP Estimation and Uses Block Matching for Homography Refinement.	Score Of 1.027 And Effectively Removes Error Motion Resulting from Accumulated Errors.		
A Fast and Robust Real-Time Surveillance Video Stitching Method [25]	The Proposed Method for Real-Time Video Stitching Uses Keyframes to Calculate Stitching Parameters Such as Pix Mapping Table, Stitching Seams, And Blending Weights. Fast Algorithms Are Designed to Detect Changes in Stitching Seams and Backgrounds to Determine Whether to Update Stitching Parameters or Recalculate Them, Achieving Robust and Automatic Improvement of Visual Quality While Maintaining Satisfactory Real-Time Performance.	Experiments Demonstrate Its Robustness to Various Changes and Good Real-Time Performance.	Stitching Parameters Updating Is Relatively Time-Consuming.	Frame Rate
Real-Time Panorama Composition for Video Surveillance Using GPU [26]	The Paper Proposes a Real-Time Panorama Composition System for Video Surveillance Using A GPU. The System Uses a Two-Step Approach That Includes Feature Matching and Image Blending to Create the Panoramic View.	The System Was Tested on A Dataset of Video Frames and Achieved a High Frame Rate Of 20 Fps Using A GPU. The Paper Concludes That the Proposed System Is Effective and Efficient for Real-Time Video Surveillance Applications.	The System Assumes a Static Camera Position, Which May Not Be Applicable in All Scenarios.	Frame Rate
Content-Preserving Video Stitching Method for Multi-Camera Systems [27]	The Paper Proposes a Content-Preserving Video Stitching Method for Multi-Camera Systems. The Proposed Method Preserves the Content of The Input Videos by Minimizing Distortion and Misalignment in The Stitched Output Video. The Method Uses a Global and Local Optimization Approach to Improve the Stitching Quality.	The Proposed Method Was Tested on A Dataset of Video Sequences and Was Compared to Several Existing Stitching Methods. The Results Showed That the Proposed Method Outperformed the Existing Methods in Terms of Content Preservation and Stitching Quality.	The Proposed Method May Be Computationally Intensive and May Require More Processing Time Compared to Some Existing Methods.	Content Preservation and Stitching Quality
Multi-Camera Video Stitching Based on Foreground Extraction [28]	The Paper Proposes a Multi-Camera Video Stitching Method Based on Foreground Extraction. The Proposed Method	The Proposed Method Was Tested on A Dataset of Video Sequences and Was Compared to Several Existing Methods. The	The Proposed Method May Be Sensitive to Changes in Foreground Objects and May Require Additional Processing	Stitching Accuracy and Processing Time

	Uses Foreground Masks to Stitch Multiple Videos Captured from Different Cameras into A Single Panoramic Video. The Method Includes Several Stages Such as Camera Calibration, Foreground Extraction, And Stitching.	Results Showed That the Proposed Method Outperformed the Existing Methods in Terms of Stitching Accuracy and Processing Time.	to Handle Dynamic Scenes.	
Real-Time Panorama and Image Stitching with Surf-Sift Features [29]	The Paper Proposes a Real-Time Panorama and Image Stitching Method Using SURF-SIFT Features. The Proposed Method Uses a Hybrid Approach That Combines SURF Features for Feature Detection and SIFT Features for Feature Matching and Stitching. The Method Includes Several Stages Such as Feature Detection, Feature Matching, And Image Stitching.	The Proposed Method Was Tested on A Dataset of Images and Was Compared to Several Existing Methods. The Results Showed That the Proposed Method Outperformed the Existing Methods in Terms of Image Quality, Stitching Accuracy, And Processing Time.	The Proposed Method May Be Sensitive to Changes in Image Lighting Conditions and May Require Additional Processing to Handle Dynamic Scenes.	Success Rate and Processing Time
Real-Time Video Stitching for Mine Surveillance Using a Hybrid Image Registration Method [30]	The Paper Proposes a Real-Time Video Stitching Method for Mine Surveillance Using a Hybrid Image Registration Method. The Proposed Method Combines a Feature-Based Approach and A Dense-Based Approach for Image Registration. The Method Includes Several Stages Such as Feature Extraction, Feature Matching, Dense Matching, And Image Blending.	The Proposed Method Was Tested on A Dataset of Videos Captured from Different Cameras in A Mine Environment. The Results Showed That the Proposed Method Achieved Real-Time Video Stitching with High Stitching Accuracy.	The Proposed Method May Require Additional Processing to Handle Dynamic Scenes and Changes in Lighting Conditions.	Processing Time, Stitching Score, Stability Score and Trajectory Score
An Efficient Multi-View Panoramic Imaging and Extra Compression of Surveillance Cameras' Footage Using Stitching [31]	The Paper Proposes an Efficient Multi-View Panoramic Imaging and Extra Compression Method for Surveillance Cameras Using Stitching. The Proposed Method Includes Several Stages Such as Feature Detection, Feature Matching, Image Stitching, And Extra Compression Using H.264/AVC Video Compression.	The Proposed Method Was Tested on A Dataset of Surveillance Camera Footage and Was Compared to Several Existing Methods. The Results Showed That the Proposed Method Outperformed the Existing Methods in Terms of Compression Efficiency While Maintaining High Image Quality.	The Proposed Method May Require Additional Processing Power and May Not Be Suitable for Real-Time Surveillance Applications.	Stitching Score and Stability Score

A Survey on Image and Video Stitching [32]	The Paper Provides a Survey of Image and Video Stitching Techniques. The Survey Includes Various Approaches Such as Feature-Based Methods, Homography-Based Methods, Graph-Based Methods, And Deep Learning-Based Methods. The Paper Also Discusses the Challenges and Applications of Image and Video Stitching.	The Paper Does Not Present Any Experimental Results or New Proposed Methods.	The Paper Does Not Present Any Experimental Results or New Proposed Methods.	The Paper Does Not Present Any Experimental Results or New Proposed Methods.
Joint Video Stitching and Stabilization from Moving Cameras [33]	The Paper Proposes a Joint Video Stitching and Stabilization Method for Moving Cameras. The Proposed Method Combines Feature-Based Stitching and Motion-Based Stabilization to Produce a Stabilized Panoramic Video.	The Proposed Method Was Tested on Several Datasets of Moving Camera Videos and Was Compared to Several Existing Methods. The Results Showed That the Proposed Method Outperformed the Existing Methods in Terms of Stitching Accuracy and Stabilization Quality.	The Proposed Method May Require High Processing Power and May Not Be Suitable for Real-Time Applications.	Processing Time, Stitching Score, Stability Score.

2.5 Object Detection

Yolov4 is a state-of-the-art real-time object detection system that achieves high accuracy and fast processing speed. It has been designed to meet the demands of modern applications such as self-driving cars, surveillance systems, and robotics, (see figure 2-7, that shows the process yolov4 take to detect objects). One of the main advantages of Yolov4 is its ability to detect objects with high accuracy while maintaining a very high processing speed. This is achieved through a number of optimizations in the architecture and the use of various techniques such as multi-scale prediction, anchor boxes, and a novel focus loss function [34].

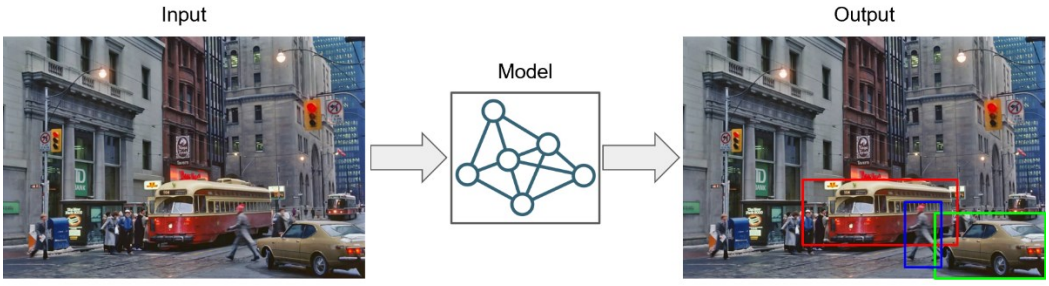

Figure 2-7 the process of object detection

In addition to its high accuracy and processing speed, Yolov4 has also been optimized for use in lightweight models. This is particularly important for applications where computational resources are limited, such as mobile devices and embedded systems. To achieve this, Yolov4 has been designed to be modular, with different layers and components that can be added or removed depending on the specific needs of the application. This allows developers to create customized models that are optimized for their specific use case, while still maintaining high accuracy and processing speed [34].

One of the limitations of Yolov4 is that it can be computationally intensive when used with high-resolution images. This can result in slower processing times and increased memory usage, which can be a challenge for applications with limited resources. To address this, various optimizations such as pruning, and quantization techniques have been proposed to reduce the computational complexity of the model without sacrificing accuracy [35].

To evaluate the performance of Yolov4, various benchmark datasets such as COCO, Pascal VOC, and KITTI have been used. The most commonly reported metrics for object detection are accuracy and mean average precision (mAP). Yolov4 has consistently achieved state-of-the-art results on these datasets, with mAP scores of over 50% on COCO and over 80% on Pascal VOC [36]. Table 2-2 summarizes part of the work done in this area.

Table 2-2 Object detection algorithms

Name	Description	Results	Limitations	Dataset
YOLOv4: Optimal Speed and Accuracy of Object Detection [36]	Trained and fused the fake score of two networks. One is a GoogleLeNet for detecting the face artifacts and the other is a patch-based triplet network with an SVM classifier to better capture the local noise residuals and camera characteristics features.	YOLOv4 outperforms previous versions of YOLO and other state-of-the-art object detection models on the COCO dataset with a mAP of 65.7% and a frame rate of 63 FPS on a Tesla V100 GPU.	YOLOv4 is computationally demanding and requires high-end GPUs for real-time performance.	The COCO dataset, KITTI (Karlsruhe Institute of Technology and Toyota Technological Institute) dataset and Open Images dataset are used for evaluation.
YOLOv4-tiny: A Real-Time Object Detection Model Optimized for Resource-Constrained Devices [35]	This paper proposes YOLOv4-tiny, a lightweight version of YOLOv4 designed for resource-constrained devices. The model achieves a balance between accuracy and speed by reducing the number of layers and channels and implementing various optimizations.	Their YOLOv4-tiny model achieves state-of-the-art performance on the COCO dataset, achieving a mAP of 57.1% and a speed of 244 FPS on an NVIDIA Jetson Nano.	The model sacrifices some accuracy to achieve real-time performance on resource-constrained devices.	The COCO dataset is used for evaluation.

Real-Time Object Detection System for Autonomous Vehicles Using YOLOv4 [34]	This paper presents a real-time object detection system for autonomous vehicles using YOLOv4. The authors optimize the YOLOv4 model for deployment on a Jetson Nano platform and show that it achieves high accuracy and real-time performance on various road scenarios.	The optimized YOLOv4 model achieves a mAP of 39.2% on the Udacity self-driving car dataset and a speed of 30 FPS on a Jetson Nano.	The system may not generalize well to other road scenarios and may require further optimization for deployment on other hardware platforms.	Udacity self-driving car

2.6 Summary

This chapter presented a review on Real-time video stitching involves combining multiple video streams captured from either static or moving cameras into a seamless panoramic video in real-time. Static camera video stitching requires the cameras to be fixed and maintaining a constant relative geometry during the stitching process. Moving camera video stitching, on the other hand, involves dealing with issues related to camera motion and maintaining consistent geometry. Both static and moving camera video stitching involve challenges such as frame alignment, color correction, and detecting and repairing blurred frames. In addition, moving camera video stitching requires addressing temporal properties such as video synchronization and video stabilization to eliminate shaky and jittery movements. Effective handling of camera motions is crucial in video stitching, and the combination of spatial and temporal artifacts can make the overall task even more challenging. However, innovative computer vision solutions are being developed to address these challenges.

Yolov4 is a powerful object detection system that offers high accuracy and fast processing speed. Its modular architecture allows for customization and optimization for specific use cases, and it has been optimized for use in lightweight models. While it can be computationally intensive with high-resolution images, various optimizations have been proposed to reduce complexity without sacrificing accuracy. Overall, Yolov4 is a versatile and effective tool for a wide range of object detection applications. An explanation for the methodologies created by previous research as well as their used dataset, performance measures and limitations were discussed in this chapter.

CHAPTER THREE
BACKGROUND

3 BACKGROUND

This chapter includes a brief introduction on the hardware used in this thesis such as Tello drones and RPI, and the operating principle of some of the used technologies/algorithms used in this thesis such as features from accelerated segment test, random sample consensus, Kanade–Lucas–Tomasi feature tracker, convolution neural network.

3.1 Tello Drones

The Tello drone is a small and affordable quadcopter that was developed by Ryze Tech in collaboration with DJI and is powered by a Qualcomm Snapdragon processor, shown in figure 3-1.

Figure 3-1 Tello drone

The Tello drone has a number of impressive hardware specifications, given its small size and price point. It can reach speeds of up to 28 km/h. The Tello drone weighs only 80 grams and has a range of 100 meters. The drone is equipped with the following:

- Infrared Sensor: This sensor helps the drone to maintain stability by measuring the distance between the drone and the ground. It also allows the drone to perform auto takeoff and landing.
- Barometer: The barometer measures changes in air pressure, which allows the drone to maintain its altitude.
- Vision Positioning System: The vision positioning system consists of a downward-facing camera and a group of sensors that allow the drone to detect and avoid

obstacles. It also helps the drone to maintain its position and hover accurately indoors.

- 5MP Camera: The Tello drone is equipped with a 5 MP camera that can capture 720p video at 30 frames per second. The camera is mounted on the front of the drone and can be adjusted manually.
- Battery: The Tello drone has a removable 3.8V 1100mAh LiPo battery that provides up to 13 minutes of flight time.
- Wi-Fi Connectivity: The Tello drone creates its own Wi-Fi network, which allows it to communicate with a smartphone or other device.

The drone is equipped with this number of sensors as they help it to maintain stability and avoid obstacles, as shown in figure 3-2.

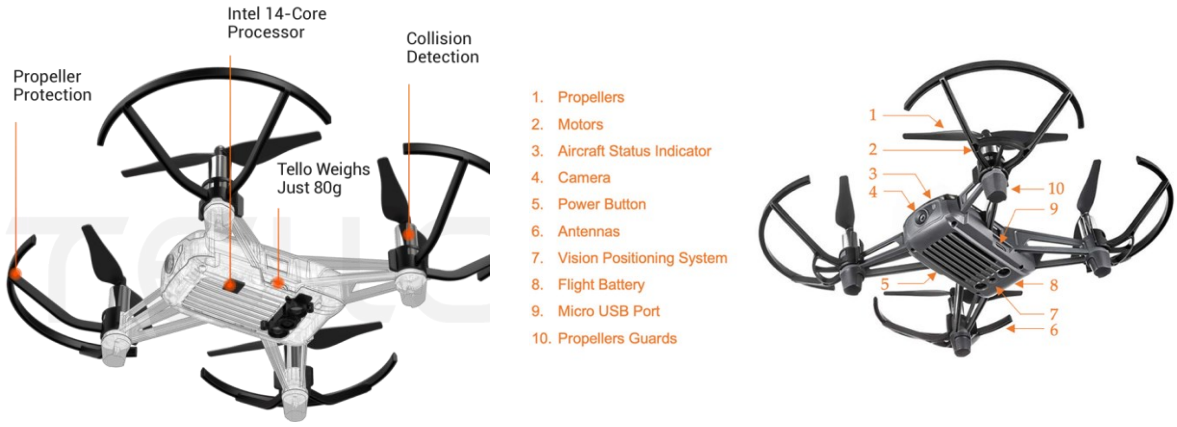

Figure 3-2 Tello drone sensors and component placement

Connecting a Tello drone to a PC using a hotspot is another popular method for integrating Tello drones into autonomous flight systems. A hotspot is a wireless network created by the Tello drone that allows other devices, such as a PC, to connect to the drone's Wi-Fi network. By connecting to the Tello drone's Wi-Fi network, the PC can send and receive commands to and from the drone, allowing it to control the drone's flight and camera functions.

To connect a Tello drone to a PC using a hotspot, first, enable the drone's hotspot mode by pressing and holding the power button until the lights start flashing rapidly. Then, connect your PC to the drone's Wi-Fi network using the Wi-Fi settings on your PC. Once the PC is connected to the drone's network, you can use the Tello SDK to send commands to the drone and receive

telemetry data from the drone. This enables you to control the drone's flight and camera functions using the Tello SDK [37], and integrate it into an autonomous flight system.

Overall, connecting a Tello drone to a PC using a hotspot is a simple and effective way to integrate Tello drones into autonomous flight systems, see figure 3-3, which illustrate the connection between Tello drone's hotspot and a computer. By leveraging the drone's Wi-Fi network and the Tello SDK. To retrieve a live video stream from the Tello drone, the Tello SDK provides a "VideoStream" object that can be used to establish a Wi-Fi connection with the drone and start receiving video data. The video stream is sent in H.264 format with a resolution of 720p at 30 frames per second, and it can be accessed through the "get_frame_read ()" method of the "VideoStream" object. This method retrieves the latest frame from the video stream as a NumPy array, which can be further processed using computer vision libraries like OpenCV.

Figure 3-3 Tello connecting to PC using Tello's hotspot

3.2 Raspberry PI

The Raspberry Pi 4 is the latest and most powerful single-board computer developed by the Raspberry Pi Foundation. The board is equipped with a powerful 64-bit quad-core ARM Cortex-A72 processor, which can run at up to 1.5GHz, and comes with up to 8GB of RAM, making it an ideal platform for running complex applications and tasks.

In addition to its powerful CPU and RAM, the Raspberry Pi 4 also comes with a range of connectivity options, including dual-band 802.11ac Wi-Fi, Gigabit Ethernet, Bluetooth 5.0, and USB 3.0 ports. It also has dual micro-HDMI ports that support up to two 4K displays at 60 frames per second.

The Raspberry Pi 4 is widely used for a variety of applications, such as robotics, home automation, media centers, and IoT devices. With its powerful CPU, high amount of RAM, and

extensive connectivity options, it provides a flexible and capable platform for developers and hobbyists alike.

Overall, the Raspberry Pi 4 is an excellent choice for those who need a powerful and versatile single-board computer that can handle complex tasks and run a wide range of applications. Its low cost, small form factor, and extensive support community make it an ideal platform for learning, experimentation, and prototyping [38].

The RPI 4, which is shown in figure 3-4, is chosen to be the base station of the drones and to do the stitching process on the received videos in real time.

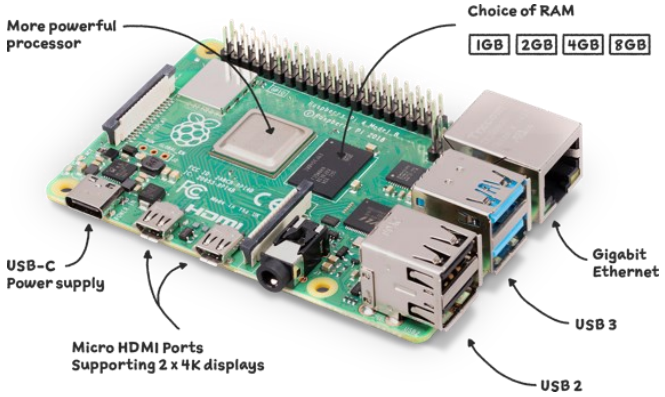

Figure 3-4 RPI 4

The Raspberry Pi Zero W is a small and affordable single-board computer that comes with built-in Wi-Fi and Bluetooth capabilities. It is based on the same processor as the Raspberry Pi 1 but has a smaller form factor and a lower power consumption.

The Raspberry Pi Zero W features a 1GHz single-core CPU, 512MB of RAM, and support for microSD cards. It also comes with a mini-HDMI port, a micro-USB OTG port, and a GPIO header. Its built-in Wi-Fi and Bluetooth capabilities make it easy to connect to the internet and other devices, without the need for additional adapters.

The Raspberry Pi Zero W is an ideal platform for a variety of projects, such as IoT devices, media centers, and robotics. Its small form factor and low power consumption make it easy to integrate into other projects or use in portable applications.

Overall, the Raspberry Pi Zero W, shown in figure 3-5, is a powerful and versatile single-board computer that offers built-in Wi-Fi and Bluetooth capabilities, making it easy to connect to the internet and other devices. Its low cost and small form factor make it an excellent choice

for a variety of applications, and its extensive support community ensures that there are plenty of resources available for users and developers alike [38].

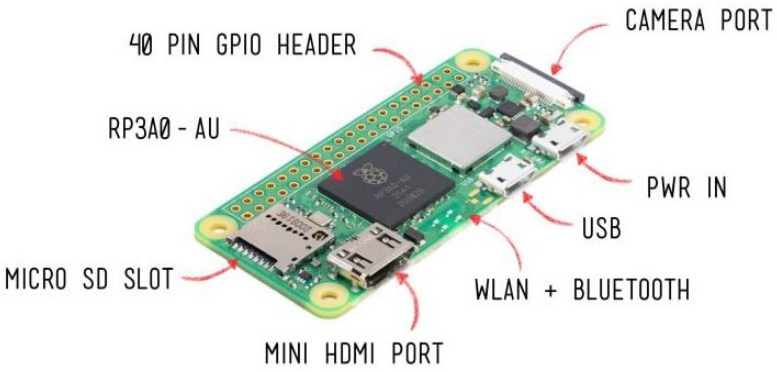

Figure 3-5 RPI Zero W

3.3 Features from Accelerated Segment Test (FAST)

The Features from Accelerated Segment Test (FAST) is a corner detection algorithm used in computer vision and image processing. It is a popular feature detection algorithm due to its computational efficiency and speed. The FAST algorithm is used to detect corners in an image, which are used as key points in feature-based computer vision applications, such as object recognition and tracking.

The FAST algorithm works by comparing the intensity of pixels in a circular pattern around a central pixel. The algorithm considers a pixel to be a corner if it has a higher intensity than a certain threshold and is surrounded by a sufficient number of contiguous pixels that are brighter or darker than the central pixel. This process is repeated for all pixels in the image, resulting in a set of corner features that can be used for further processing.

FAST has several features that make it a popular algorithm for corner detection. One of its primary advantages is its computational efficiency, which allows it to operate in real-time on video streams and large images. FAST is also invariant to rotation and scale, which makes it robust to changes in the size and orientation of objects in an image. Additionally, FAST is relatively simple to implement and can be used in combination with other feature detection algorithms, such as SIFT (Scale-Invariant Feature Transform) and SURF (Speeded-Up Robust Features), to improve their performance.

The choice of a circle with a radius of 3 pixels is a good compromise between accuracy and speed, making FAST a popular choice for real-time applications where computational efficiency is a priority, such as feature tracking in video streams or robotic vision systems.

The mathematical equations for the FAST algorithm are as follows:

Let p be the intensity of the central pixel and t be the threshold value.
Let c be the number of contiguous pixels needed to define a corner.
Case 1: p is brighter than its surrounding pixels.
Let I (x, y) be the intensity of the pixel at location (x, y).
A pixel (x, y) is a corner if:
I (x, y) > p + t and there are c contiguous pixels in a circle of radius 3 around (x, y) that are brighter than p.
Case 2: p is darker than its surrounding pixels.
A pixel (x, y) is a corner if:
I (x, y) < p - t and there are c contiguous pixels in a circle of radius 3 around (x, y) that are darker than p.
Case 3: p is in between its surrounding pixels.
A pixel (x, y) is a corner if:
I (x, y) > p + t and there are c contiguous pixels in a circle of radius 3 around (x, y) that are brighter than p, or
I (x, y) < p - t and there are c contiguous pixels in a circle of radius 3 around (x,y) that are darker than p.

See figure 3-6, which shows 12-point segment test corner detection in an image patch. The highlighted squares are the pixels used in the corner detection. The pixel at p is the center of a candidate corner. The arc is indicated by the dashed line passes through 12 contiguous pixels which are brighter than p by more than the threshold.

Here's an example Pseudo code for implementing the FAST algorithm [39]:

1. Input the grayscale image.

2. For each pixel in the image:
    - Compute the score for the pixel as the difference in intensity values between the pixel and the intensity values of the surrounding pixels in a circle of radius 3 pixels.
    - If the pixel score is above a threshold and is an extremum (i.e., brighter, or darker than all other pixels within a circle of radius 3 pixels), mark it as a corner feature.
3. Output the corner features.

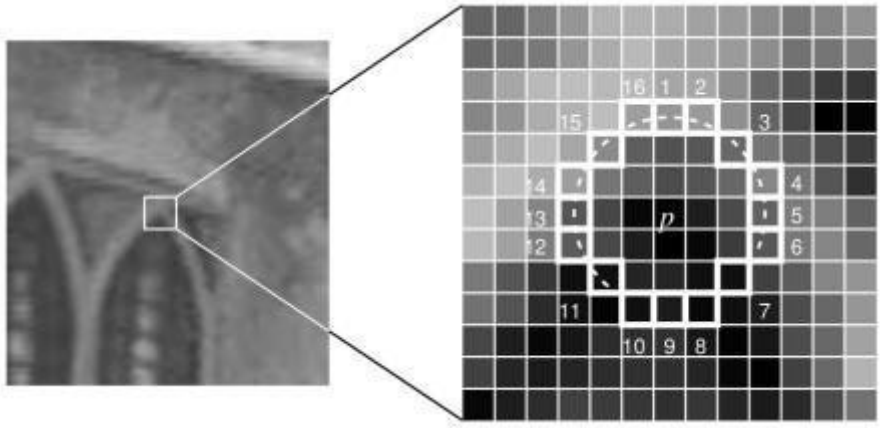

Figure 3-6 12-point segment test corner detection in an image patch

Overall, the FAST algorithm is a powerful and efficient corner detection algorithm that is widely used in computer vision and image processing applications. Its ability to operate in real-time and its robustness to rotation and scale make it an ideal choice for feature-based applications, such as object recognition and tracking [40].

3.4 Kanade–Lucas–Tomasi feature tracker

The Kanade-Lucas-Tomasi (KLT) feature tracker is a widely used algorithm for tracking features in a sequence of images. It is an improvement over the Lucas-Kanade algorithm, which only tracks a single feature, by tracking multiple features in an image.

The KLT algorithm works by selecting a set of features, called keypoints, in an initial image. The algorithm then tracks these keypoints in subsequent images by computing the optical flow, which is the apparent motion of the keypoints between images. The optical flow is computed using a local search window around each keypoint and minimizing the sum of squared differences between the pixel intensities in the search window in the current and previous frames [41].

The mathematical equations for the KLT algorithm are as follows:

Let I(x, y, t) be the intensity of the pixel at location (x,y) and time t.
Let u and v be the horizontal and vertical displacement of the keypoint between frames.
The optical flow equation can be written as:
$\sum(I(x,y,t) - I(x+u, y+v, t+1))^2 = \min$

This equation can be solved using the Lucas-Kanade algorithm, which involves computing the gradient of the image with respect to x and y and the temporal derivative of the image. The optical flow is then computed by solving a system of linear equations.

Here's an example Pseudo code for implementing the KLT algorithm using OpenCV:

1. Input the current frame and the reference frame.
2. Detect features in the reference frame using a feature detector (e.g., Harris corner detector)
3. For each feature in the reference frame, compute the optical flow vector using Lucas-Kanade algorithm:
   - Create a small window around the feature in the reference frame.
   - Compute the gradient of the image in the window.
   - Compute the optical flow vector that minimizes the sum of squared differences between the reference window and the corresponding window in the current frame.
4. Discard features with large optical flow vectors or low corner response.
5. Output the remaining features and their optical flow vectors.

3.5 Random Sample Consensus

The RANSAC (Random Sample Consensus) algorithm is an iterative method for estimating the parameters of a mathematical model from a set of observed data that contains outliers. It is commonly used in computer vision for tasks such as robust estimation of geometric transformations or fitting of 3D models to point clouds.

The main idea of the RANSAC algorithm is to randomly select a subset of the data (called inliers) and estimate the model parameters using only this subset. Then, the algorithm tests the model on the remaining data (called outliers) and counts the number of data points that are consistent with the model within a given tolerance. This process is repeated many times with

different random subsets, and the subset that produces the best fit (i.e., the highest number of inliers) is chosen as the final model [42].

The RANSAC algorithm can be summarized in the following steps:

Data: Number of inliers we use each iteration n, maximum number of iterations m, input data *data*, threshold for determining a good fit t, number of close points required for a good model fit k
Result: Model *bestFit*
$i \leftarrow 0$;
$bestFit \leftarrow null$;
$bestError \leftarrow infinite$;
while $i < m$ **do**
  $tempInliers \leftarrow \text{selectRandom}(data, n)$;
  $tempInliersAdd \leftarrow empty$;
  $tempModel \leftarrow \text{fit}(tempInliers)$;
  for *point in data* **do**
    if *point not in tempInliers* **then**
      if $distance(point, tempModel) < t$ **then**
        $tempInlierAdd.\text{add}(point)$;
      end
    end
  end
  if $length(tempInliersAdd) > k$ **then**
    $newModel \leftarrow \text{fit}(maybeInliers \text{ and } tempInliers)$;
    if $newModel.error > bestError$ **then**
      $bestError \leftarrow error$;
      $bestFit \leftarrow betterModel$;
    end
  end
end

Figure 3-7 shows that the linear regression fails to describe our data correctly. As the linear regression needs to go through the inliers only to get the data right [43].

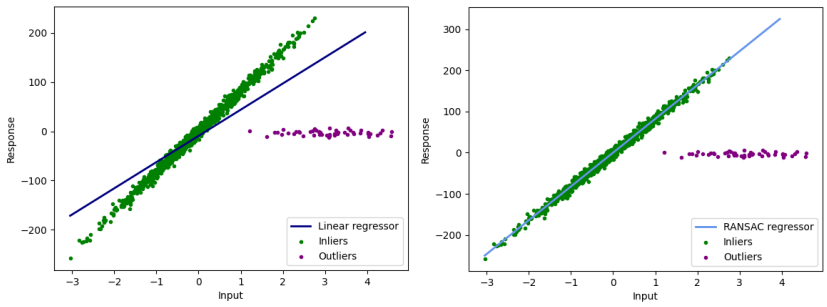

Figure 3-7 on the left: linear regression, on the right RANSAC

The higher the number of iterations, the higher the probability that we detect a subset without any outliers in it. We can use a result from statistics, that uses the ratio of inliers to total points $w = \frac{inliers}{total\ points}$, the number of data points n we need for our model calculation and the probability p that we encounter a subset without outliers in it:

$$k = \frac{\log(1-p)}{\log(1-w^n)}$$

3.6 You Only Look Once v4

One-stage and two-stage object detection algorithms are the two main approaches used for object detection tasks [36].

One-stage detectors, such as YOLO and SSD, directly predict the bounding boxes and class probabilities in a single pass through the network. They typically have faster inference times compared to two-stage detectors, since they do not require a separate region proposal stage. However, one-stage detectors often have lower accuracy compared to two-stage detectors.

Two-stage detectors, such as Faster R-CNN, first generate region proposals (bounding box candidates) in a separate stage before predicting the class probabilities and refining the bounding box coordinates in a second stage. They usually have slower inference times compared to one-stage detectors, but often have higher accuracy due to the separate region proposal stage and the ability to refine the bounding box coordinates more accurately [36].

In general, one-stage detectors are better suited for real-time applications where speed is critical, such as robotics, autonomous driving, and video analysis. On the other hand, two-stage detectors are often used in applications where high accuracy is more important, such as medical imaging, object tracking, and satellite imaging [36].

It is worth noting that the performance of one-stage and two-stage detectors can be influenced by a variety of factors, such as the choice of network architecture, the training data, and the specific application domain. Therefore, it is important to carefully consider the trade-offs between speed and accuracy when selecting an object detection algorithm for a particular task.

Object detectors utilize convolutional neural network backbones to extract features from an input image, which are then combined in the neck to enable the drawing of multiple bounding boxes around objects and their classification. In contrast to image classification, object detection requires the mixing of multiple feature layers. Object detectors can be divided into two

categories: one-stage detectors and two-stage detectors, depending on whether object localization and classification are decoupled or predicted simultaneously for each bounding box [36], see figure 3-8, which explains the one stage detector and two stage detectors.

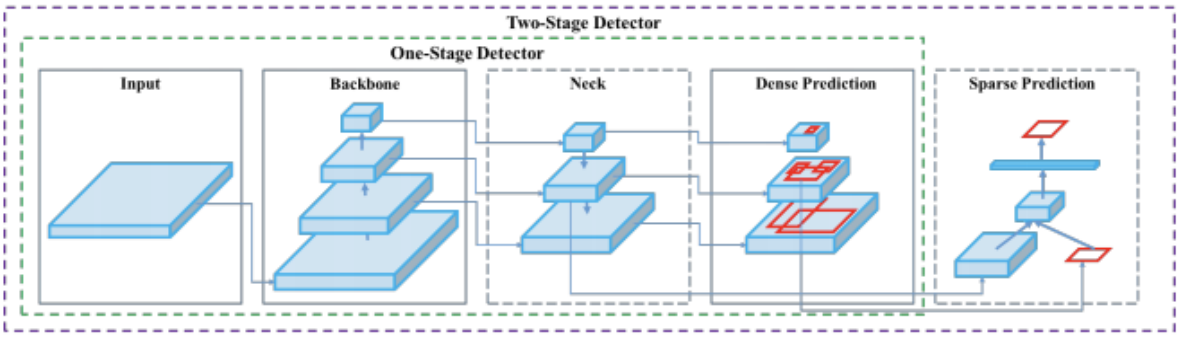

Figure 3-8 The Anatomy of an Object Detector

YOLOv4 is a deep neural network model that performs object detection in real-time. It works by dividing the input image into a grid of cells and predicting bounding boxes and class probabilities for each cell. This approach allows YOLOv4 to detect multiple objects in a single pass and with high accuracy [44].

YOLOv4 is a one-stage object detection algorithm, which means that it directly predicts the bounding boxes and class probabilities in a single pass through the network. In contrast, two-stage object detection algorithms, such as Faster R-CNN, first generate region proposals (bounding box candidates) in a separate stage before predicting the class probabilities and refining the bounding box coordinates in a second stage [36].

One-stage object detectors like YOLOv4 typically have faster inference times compared to two-stage detectors, since they do not require a separate region proposal stage. However, two-stage detectors often have higher accuracy because they can refine the bounding box coordinates more accurately.

To improve the accuracy of YOLOv4, the authors of the paper introduced a number of new techniques, such as the SPP-block, the PANet, and the Mish activation function. These techniques help to improve the accuracy of the bounding box predictions and reduce false positives, which are common problems in one-stage detectors.

The backbone network of an object detector is usually pre-trained on the ImageNet dataset for image classification. This pre-training enables the network to identify relevant features in an image, which can then be adapted to the new task of object detection through fine-tuning.

The YOLOv4 object detector considered several backbone networks, including CSPResNext50, CSPDarknet53, and EfficientNet-B3. These backbones were chosen for their ability to extract meaningful features from images and their efficiency in terms of computational complexity and memory usage.

CSPResNext50 is a modified version of the ResNeXt architecture that uses cross-stage partial connections to improve feature representation and reduce computation. CSPDarknet53 is a modified version of the Darknet architecture used in previous versions of YOLO, which also uses cross-stage partial connections to improve feature representation. EfficientNet-B3 is a neural network architecture designed to achieve high accuracy with low computational cost by balancing the number of parameters and computational complexity.

The authors of YOLOv4 experimented with different combinations of these backbones and found that CSPDarknet53 provided the best balance between accuracy and computational efficiency, making it the backbone of choice for YOLOv4. However, they also noted that CSPResNext50 and EfficientNet-B3 could be viable alternatives for certain use cases [36], see figure 3-9, that illustrates the parameters of the three backbones of YOLOv4.

Table 3-1 Parameters of neural networks for image classification

Backbone model	Input network resolution	Receptive field size	Parameters	Average size of layer output (WxHxC)	BFLOPs (512x512 network resolution)	FPS (GPU RTX 2070)
CSPResNext50	512x512	425x425	20.6 M	**1058 K**	31 (15.5 FMA)	62
CSPDarknet53	512x512	725x725	**27.6 M**	950 K	52 (26.0 FMA)	**66**
EfficientNet-B3 (ours)	512x512	**1311x1311**	12.0 M	668 K	11 (5.5 FMA)	26

Both CSPResNext50 and CSPDarknet53 are based on DenseNet, a type of convolutional neural network designed to alleviate the vanishing gradient problem, improve feature propagation, encourage feature reuse, and reduce the number of parameters in the network. The motivation behind DenseNet was to address the difficulty of backpropagating loss signals through deep neural networks. By connecting the layers in a dense manner, DenseNet enables better information flow between layers and facilitates feature reuse, ultimately resulting in more efficient and effective neural network training. CSPResNext50 and CSPDarknet53 build upon

the DenseNet architecture and incorporate additional features, such as cross-stage partial connections, to further enhance their performance in object detection tasks [45], see figure 3-10 illustrates the layout of the resulting DenseNet schematically.

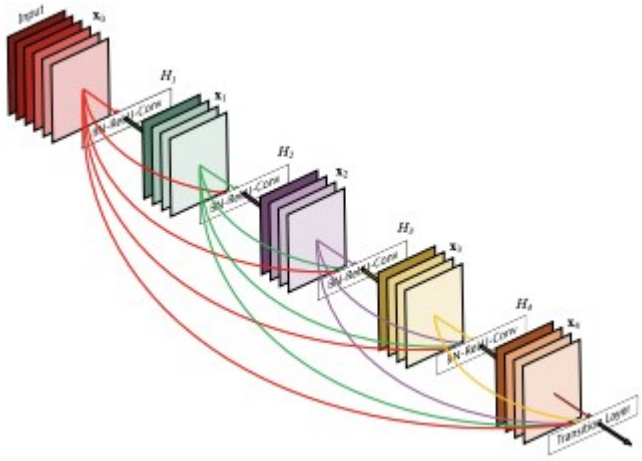

Figure 3-9 A 5-layer dense block with a growth rate of k = 4,.

After extracting features from the ConvNet backbone, the next step in object detection is to mix and combine these features in the neck to prepare for the detection process. YOLOv4 offers several options for the neck, including FPN, PAN, NAS-FPN, BiFPN, ASFF, and SFAM.

The neck components typically flow up and down among layers and connect only the few layers at the end of the convolutional network. FPN, or Feature Pyramid Network, is a popular choice that creates a pyramid of multi-scale feature maps and combines them through a top-down and bottom-up pathway to provide scale-invariant object detection. PAN, or Path Aggregation Network, is a modification of FPN that combines feature maps from different scales in a parallel and adaptive manner.

NAS-FPN, or Neural Architecture Search FPN, uses a neural architecture search approach to automatically generate a feature pyramid network optimized for object detection. BiFPN, or Bidirectional Feature Pyramid Network, is a modification of FPN that incorporates top-down and bottom-up pathways in a bidirectional manner to improve information flow between layers.

ASFF, or Adaptive Spatial Feature Fusion, is a method for adaptive feature fusion that dynamically adjusts the weight of feature maps based on their importance for object detection. SFAM, or Spatial Attention Module, is a self-attention mechanism that learns to selectively attend to important spatial locations in the feature maps.

The choice of neck architecture depends on the specific requirements of the object detection task, such as the desired level of accuracy, computational complexity, and memory usage, see figure 3-11 shows the conventional top-down approach for different neck architecture [46].

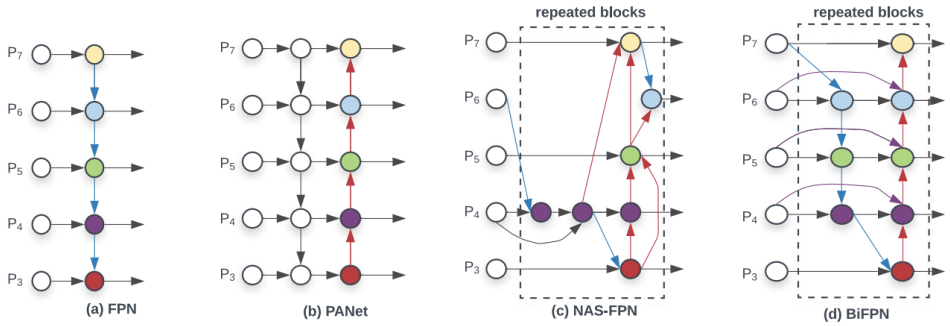

Figure 3-10 Feature network design – (a) FPN introduces a top-down pathway to fuse multi-scale features from level 3 to 7 (P3 - P7); (b) PANet adds an additional bottom-up pathway on top of FPN; (c) NAS-FPN use neural architecture search to find an irregular feature network topology and then repeatedly apply the same block; (d) is our BiFPN with better accuracy and efficiency trade-offs

Each one of the P_i above represents a feature layer in the CSPDarknet53 backbone. YOLOv4 chooses PANet as the feature aggregation method in the neck of the network, PANet has shown to be an effective method for combining multi-scale feature maps in object detection tasks.

As for the SPP block, or Spatial Pyramid Pooling block, it is added after the CSPDarknet53 backbone in YOLOv4 to increase the receptive field and capture important features across multiple scales. The SPP block applies multiple pooling operations at different scales and concatenates the resulting features to produce a fixed-length representation of the input image, which is then fed into the detection head. This helps to separate out the most important features from the backbone and improve the accuracy of object detection [46].

YOLOv4 uses the same detection head as YOLOv3, which is based on anchor boxes for object localization. The detection head takes the feature maps produced by the neck and applies a series of convolutional layers to predict the class probabilities, confidence scores, and bounding box offsets for each anchor box.

YOLOv4 also employs three levels of detection granularity, which means that the detection head outputs predictions at three different scales, each corresponding to a different level of feature map in the neck. The use of multiple scales helps to improve the detection accuracy of objects at different sizes and distances from the camera. The predicted boxes at different scales

are then processed using non-maximum suppression to remove duplicate detections and produce the final set of object detections for the image, see figure 3-12 which illustrates the bounding box offsets for each anchor box.

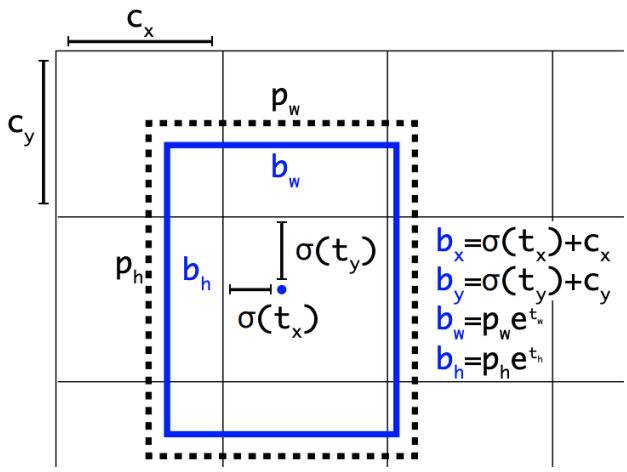

Figure 3-11 Bounding boxes with dimension priors and location

YOLOv4 was extensively evaluated on the MS COCO dataset, which is a widely used benchmark for object detection algorithms. The COCO dataset consists of images from a wide range of scenarios and includes 80 different object classes, making it a good test for the generalization capabilities of an object detector [36].

YOLOv4 incorporates a range of state-of-the-art techniques for object detection, such as a CSP backbone, a PANet feature aggregation network, and a SPP block to increase the receptive field. These techniques have been tested and refined to create a highly accurate and efficient real-time object detector.

One of the key benefits of YOLOv4 is its ease of use and versatility. It can be easily trained on custom objects using a GPU or cloud-based services like Google Colab. This makes it a popular choice for a wide range of computer vision applications, from security and surveillance to robotics and self-driving cars.

3.8 Summary

This chapter presented a brief introduction of the hardware and operating principles for some of the basic concepts used in this thesis such as Tello drones and RPI, and the operating principle of some of the used technologies/algorithms used in this thesis such as features from accelerated

segment test, random sample consensus, Kanade–Lucas–Tomasi feature tracker, convolution neural network. Alongside the reason why each technology is introduced and its applications. The following chapter presents the proposed model which includes receiving the video frames then the stitching process and finally the object detection part.

CHAPTER FOUR
PROPOSED SYSTEM

4 PROPOSED SYSTEM

This chapter illustrates the proposed Natural-Inspired Drone Swarm Processing FOV for Efficient Multi-view Monitoring and Object Detection will be introduced. Each stage of the proposed model will be described in a separate section with a detailed illustration of their outcomes.

4.1 The Proposed System Diagram

The proposed model includes 4 main phases:

1. Receiving of live streams among Drones, in receiving live streams among drones, a solution for connecting drones to same base station is proposed as each drone is connected to through its own hotspot as will be shown in the following section.
2. Video frame processing, in video frame processing, different techniques are used, to adjust both temporal and spatial resolutions.
3. Stitching phase and panoramic construction phase, In the stitching phase, different algorithms are used to ensure smooth and artifacts free stitched output.
4. Object detection phase, as shown in figure 4-1, In the final phase, an object detection algorithm is proposed using YOLOv4, for its computational speed and lightweight. The details of each stage will be explained in the following section.

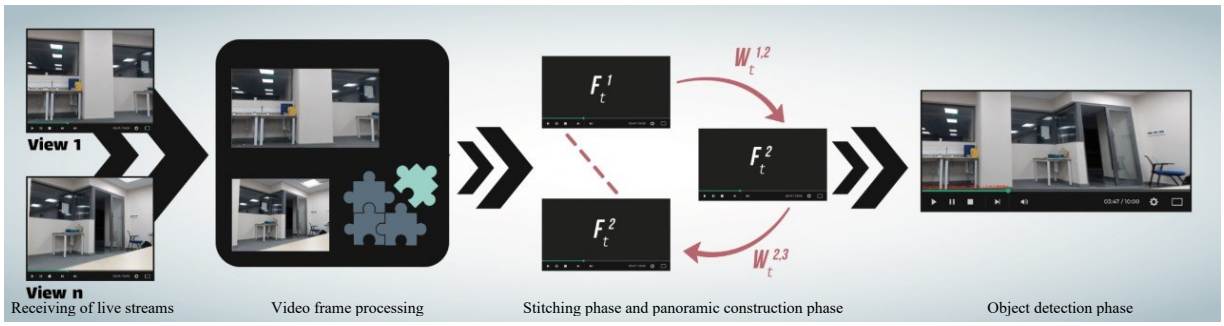

Figure 4-1 Proposed Model Diagram.

4.2 The Proposed Detection Model

The four stages illustrated in figure 4-1 will be explained thoroughly in the following subsections.

4.2.1 Receiving Video Frames

The Tello drone is a popular and affordable option equipped with a high-definition camera. The process of receiving live video streams from multiple Tello drones using a base station is described as follows, the first step is to establish a communication channel between each Tello drone and the base station. This is achieved by connecting the drones to the base station through a wireless network. Once the connection is established, the base station can request a video stream from each drone. The video stream is sent over a wireless connection as an encoded video stream. The encoded video stream must then be decoded by the base station to obtain the individual frames. This process is essential to display the live video stream from each drone. Finally, the base station can display the video streams from each drone in separate windows.

The Tello quadcopter receives commands via the Wi-Fi hotspot it emits. To control many Tello drones in a swarm, a connection to each drone's Wi-Fi hotspot is made to be able to transmit commands to each drone individually. To do so a Wi-Fi interface for each drone is used to operate it.

The methodology followed on this framework is to first start controlling the drones and receive the live streams from the multiple quadcopters, the N drones must follow a constant move or a certain degree of freedom movements, as each two adjacent drones must have overlapping areas so that the stitching process can be done. Then the starting point for stitching is initialized, as the videos must be synchronized, and since live streams are the case here, it’s only the matter of starting the stitching process while the drones are in position. It’s allowed to have fractions of one second in both static and dynamic cases but it’s not preferable.

The two drones communicate with the home station (Ground station for now), as the master node or swarm leader, which will be an equipped quadcopter that will be able to process the data and feed coming to it. Each drone transmits its video live feed and both videos are used to create the panorama wide FOV which will be used to real time detection and track objects.

All of the Tello drones have the same IP and UDP port for commanding and receiving live streams, so changing the IP addresses was a huge problem, that was solved using port rerouting method is introduced so that the receiving of multiple feed from multiple drones is available on the same device, also as all the drones are connected using the drone’s hotspot which means we have to use a network adaptor for each drone. For achieving both controlling a swarm and retrieving the video feed from the drones, each drone is connected to a raspberry pi and a WIFI

adapter to be able to connect to the drone and also have a connection to the access point so that the videos can be retrieved and the commands for the drones can be sent, see figure 4-2.

The whole system is managed using a raspberry PI 4 8G which receives the videos and starts the stitching process, the redirecting of the video streams was done as follows:

Raspberry Pi IP Address: 192.168.1.120

Port Where Video Feed is Received: 11111

Port Where Video Feed is Changed to 11117

Code for implementing the redirecting:

sudo iptables -t nat -A PREROUTING -s 192.168.1.120 -p udp --dport 11111 -j REDIRECT -- to-port 11117

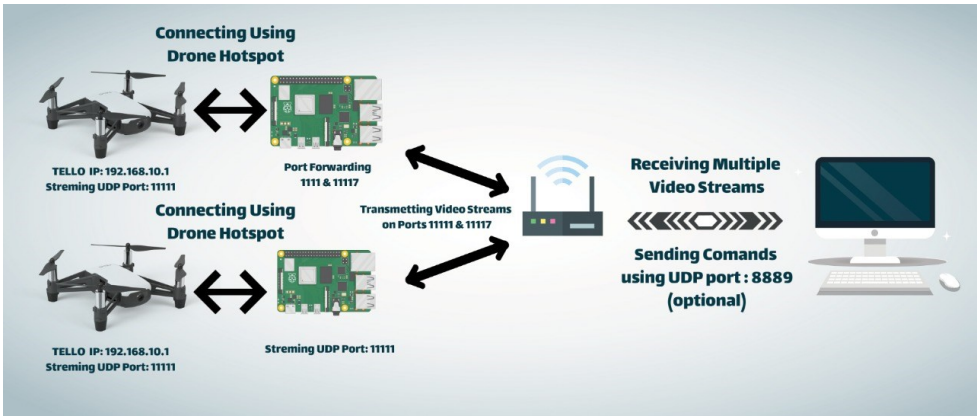

Figure 4-2 Port forwarding to receive multiple video streams at same time

4.2.2 Video Frame Processing

The pre-processing stage in real-time video stitching is a critical step that prepares the video streams for stitching. This stage starts after the video streams have been captured and must be performed before the actual video stitching process begins. The objective of the pre-processing stage is to ensure that all video streams have consistent spatial and temporal resolutions to allow for seamless and accurate stitching.

Spatial resolution refers to the size of the video frames and can be represented mathematically as the number of pixels in each frame (W x H), see figure 4-3. Temporal resolution, on the other hand, refers to the number of frames per second and can be represented mathematically as the frame rate (FPS), see figure 4-4. When different drones or cameras are used to capture the video

streams, they may return different resolutions. To unify these resolutions, the smallest spatial resolution, and the slowest frame rate of all the video streams must be chosen as the standard.

To achieve this, the following steps can be taken:

4.2.2.1 Spatial Resolution

Let (W1 x H1), (W2 x H2), ..., (Wn x Hn) be the spatial resolutions of the n video streams. The smallest resolution among these can be represented mathematically as (Wmin x Hmin) where Wmin = min (W1, W2, ..., Wn) and Hmin = min (H1, H2, ..., Hn). All other video streams must then be resized to match this standard resolution by using appropriate resizing algorithms such as bilinear interpolation or bicubic interpolation.

For moving camera scenarios Bilinear Interpolation is used, as it uses linear interpolation to estimate the value of a new pixel based on the values of its surrounding pixels. The basic idea behind Bilinear Interpolation is to use the four nearest pixels to a new pixel to estimate its value. The equation for Bilinear Interpolation can be expressed as:

$$f(x,y) = (1-\alpha)\cdot(1-\beta)\cdot f(x1,y1) + \alpha\cdot(1-\beta)\cdot f(x2,y1) \\ +(1-\alpha)\cdot\beta\cdot f(x1,y2) + \alpha\cdot\beta\cdot f(x2,y2)$$

Where $f(x, y)$ is the estimated value of the new pixel, $f(x_1, y_1)$, $f(x2, y1)$, $f(x_1, y_2)$, and $f(x_2, y_2)$ are the values of the four nearest pixels, and α and β are interpolation coefficients determined by the fractional distances between the new pixel and the nearest pixels.

For static camera scenarios Bilinear Interpolation is used, as it uses a cubic polynomial to estimate the value of a new pixel based on the values of the surrounding pixels. Unlike Bilinear Interpolation, Bicubic Interpolation uses a 16-pixel neighborhood to estimate the value of a new pixel, which provides a more accurate representation of the image. The equation for Bicubic Interpolation can be expressed as:

$$f(x,y) = \sum_{i=-1}^{2}\sum_{j=-1}^{2} a_{i,j}\cdot f(x_0+i, y_0+j)$$

Where $f(x, y)$ is the estimated value of the new pixel, $f(x_0+i, y_0+j)$ are the values of the 16 surrounding pixels, and $a_{i,j}$ are interpolation coefficients determined by the fractional distances between the new pixel and the surrounding pixels. The coefficients $a_{i,j}$ can be calculated using a set of pre-determined cubic polynomials.

In summary, Bilinear Interpolation uses linear interpolation to estimate the value of a new pixel based on the values of its surrounding pixels, while Bicubic Interpolation uses a cubic polynomial to estimate the value of a new pixel based on the values of a larger number of surrounding pixels. Bicubic Interpolation provides higher quality results compared to Bilinear Interpolation but is also more computationally intensive.

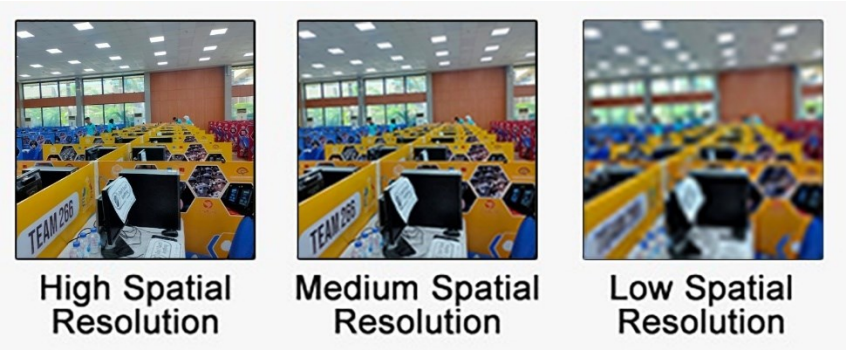

Figure 4-3 Spatial resolution examples

4.2.2.2 Temporal Resolution

Let FPS1, FPS2, ..., FPSn be the frame rates of the n video streams. The slowest frame rate among these can be represented mathematically as FPSmin where FPSmin = min (FPS1, FPS2, ..., FPSn). All other video streams must then be resampled to match this standard frame rate by using appropriate resampling algorithms such as linear interpolation or spline interpolation.

For moving camera scenarios Linear Interpolation is used, as it uses a straight line to estimate the value of a new pixel based on the values of its two nearest pixels. The equation for Linear Interpolation can be expressed as:

$$f(x,y) = (1-\alpha)\cdot f(x1,y) + \alpha \cdot f(x2,y)$$

Where $f(x,y)$ is the estimated value of the new pixel, $f(x_1,y)$ and $f(x_2,y)$ are the values of the two nearest pixels, and α is an interpolation coefficient determined by the fractional distance between the new pixel and the nearest pixels.

For static camera scenarios Spline Interpolation is used, as it uses a smooth curve to estimate the value of a new pixel based on the values of its surrounding pixels. The basic idea behind Spline Interpolation is to fit a smooth curve through the surrounding pixels and use this curve to estimate the value of the new pixel. The equation for Spline Interpolation can be expressed as:

$$f(x,y) = \sum_{i=0}^{n} a_i \cdot \varphi_i(x,y)$$

Where f(x,y) is the estimated value of the new pixel, a_i are interpolation coefficients determined by the values of the surrounding pixels, and $\varphi_i(x,y)$ are a set of pre-determined smooth functions.

In summary, Linear Interpolation uses a straight line to estimate the value of a new pixel based on the values of its two nearest pixels, while Spline Interpolation uses a smooth curve to estimate the value of a new pixel based on the values of its surrounding pixels. Spline Interpolation provides higher quality results compared to Linear Interpolation but is also more computationally intensive.

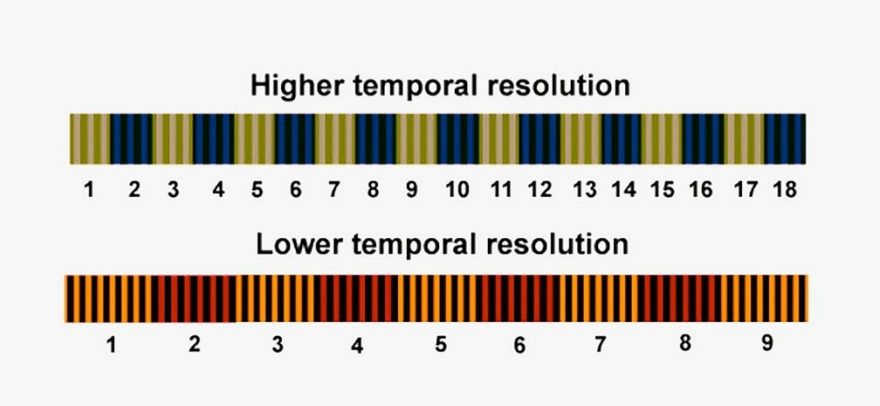

Figure 4-4 Temporal resolution examples

The pre-processing stage of real-time video stitching involves various mathematical techniques that are utilized to harmonize the spatial and temporal resolutions of multiple video streams. By performing this, it is possible to seamlessly stitch together the different video streams into a coherent and visually appealing final product. This pre-processing step is essential for achieving high-quality results in real-time video stitching applications, as it helps to overcome the challenges that arise due to the varying resolutions and frame rates of the input video streams. The ultimate goal of this stage is to ensure that the final stitched video is free from any artifacts or distortions that may detract from its overall quality.

4.2.3 Stitching and Panoramic Construction Phase

Real-time video stitching is a process that involves combining multiple video streams captured by different cameras into a single, seamless video output in real-time. This process is used in various applications such as live broadcasting, virtual reality, and surveillance systems. Real-time video stitching typically involves two main phases: the registration phase and the fusion phase. The registration phase aligns the input video streams using techniques such as feature matching and homography estimation, while the fusion phase blends the input video streams into a single output using techniques such as blending and cross-dissolving. Real-time video stitching requires sophisticated algorithms and processing techniques to achieve accurate and visually pleasing video output in real-time, see figure 4-5.

4.2.3.1 Registration Phase

Stitching registration techniques can be classified into three main approaches: direct, fast, and feature based. The direct approach involves finding correlation parameters between pixels in different images, by minimizing pixel-to-pixel dissimilarities. This method has a polynomial time complexity with respect to the number of pixels, N.

The Fast approach is designed for mobile devices with limited storage and processing power, resulting in lower quality panoramic videos. In contrast, the Brown and Lowe Method, also known as the BLM approach, introduced stitching with invariant features, which is considered the most efficient and high-quality method. This approach involves polynomial time complexity based on the number of extracted features, denoted as n, where n is significantly smaller than the total number of pixels N. The feature-based stitching algorithms typically include two primary steps: registration and blending/fusion. During the registration phase, the algorithms extract and match features.

4.2.3.2 Fusion Phase

The fusion phase, unlike the registration phase, is a more straightforward procedure that combines the images and utilizes blending techniques to create a seamless stitch. Blending methods can vary, with one involving a weighted average between the two frames. This blending technique is most effective when the image pixels are well-matched, and the disparities between the frames are mainly related to lighting. Another commonly used blending method involves

merging the images at different frequency levels, applying suitable filtering. Lower frequency levels produce softer borders, leading to a smoother blend.

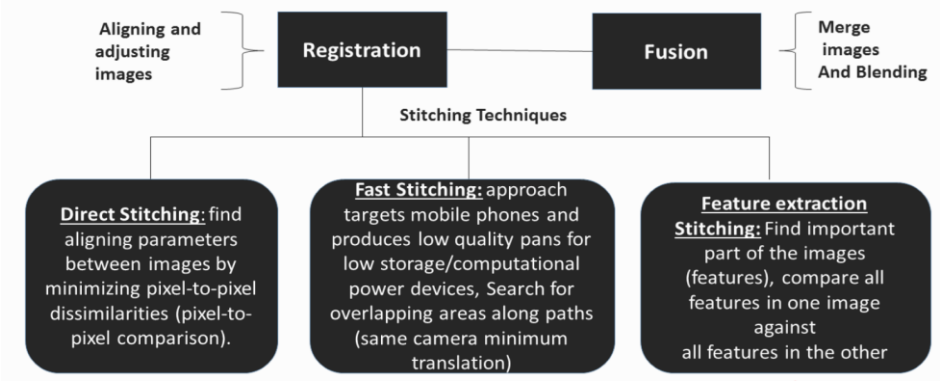

Figure 4-5 The stitching involves image registration and image fusion

The most unique process in the framework which is the estimation of two types of motion not only one, but the common use is also to estimate the motions at the corresponding frames between the multiple different videos, which is referred to by inter motions, and estimate the motions within the same video between neighboring frames, which is referred to as intra motions.

For inter motions, it can be represented as "Tn, m(t)", where 't' is the time of motion frame taken between two footages 'n' and 'm'. The mathematical equation for this can be expressed as:

$$Tn, m(t) = f(n(t), m(t))$$

Where f(n(t), m(t)) is a function that calculates the inter motion between two footages n and m at time t.

For intra motions, it can be represented as "Cn(t)", where 't' is the time and 'n' denotes the view number. The mathematical equation for this can be expressed as:

$$Cn(t) = g\big(n(t), n(t+1)\big)$$

Where g(n(t), n(t+1)) is a function that calculates the intra motion between two neighboring frames n at time t and t+1.

With two major types for path estimation which are a global path vs bundled paths, it's found that bundled paths can reduce and handle jitters, shaky frames and ghosting, which appears because of the parallax which makes global path homography model insufficient, which means some regions in the given frames won't be stabilized right, the bundled path approach is adopted in this framework as its proved to handle all the previous cases as it produces a

comparable results to the 3D methods but with respect to the metrics of the 2D methods. As the optimization stage is the core and most important stage in the whole framework its divided into three main components: the first component is to provide the best quality feature extraction on the inter motions and intra motions level, which is fast and rich feature tracking. The second component ensures the generation of the optimal camera path which is perfectly positioned in along all of the original paths to get over the perspective distortions, which is mutually optimal camera path generation. The last component is to use the generated optimal path from the previous stage to start the joint stitching and stabilization processes.

The feature detection and tracking are one the most important phases to apply the stitching algorithm, features from accelerated segment test (FAST), which uses a 16-pixels circle numbered from 1 to 16 to identify whether point P is a corner, see figure 4-6.

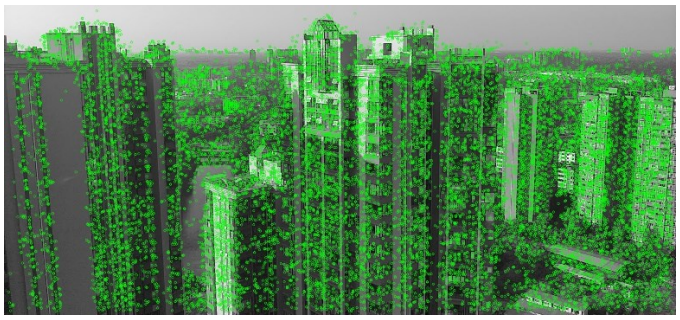

Figure 4-6 Feature detection from input frames

Final step includes point or not, and Kanade–Lucas–Tomasi (KLT) feature tracker, makes use of spatial intensity information to direct the search for the position that yields the best match, both are the fastest algorithms to be used for feature detection and tracking, which are used as this is a real time framework. Also, random sample consensus (RANSAC), which can be described as an outlier's detector or as an iterative method to estimate parameters of a mathematical model from a set of observed data that contains outliers.

Grid-based detection method is used instead of the traditional global threshold method, as the global threshold produces few features in low gradient areas, such as the sky, as the threshold is looking after the highly textured areas, that's why grid-based is much better as it gives a local value to each part of the grid so that more features can be produced, FAST is applied on each grid, and then an automatically chosen value for the grid threshold will be assigned and it will update until it reaches the best fit value. If one of the grids produced too many features, then a

pruning algorithm is introduced based on the feature detection score of each grid. After the feature detection and setting the local threshold, KLT tracker starts its work on the given frames.

In the feature detection and tracking phase, the FAST (Features from Accelerated Segment Test) algorithm is used to detect corners. The mathematical representation of FAST algorithm can be expressed as:

$$FAST(P) = \{1, 2, 3, \ldots, 16\}$$

Where P is a point in the image, and the set {1, 2, 3, ..., 16} represents the 16 pixels in a circle around the point P. The algorithm determines whether the point P is a corner based on the intensity values of these 16 pixels.

The KLT (Kanade-Lucas-Tomasi) feature tracker uses spatial intensity information to track features in the video. The mathematical representation of KLT algorithm can be expressed as:

$$KLT(I) = argmin\, ||I(x + u) - T(x)||$$

Where I(x) is the intensity of the current frame at position x, T(x) is the intensity of the previous frame at position x, and u is the displacement vector between the two frames. The KLT algorithm finds the displacement vector that minimizes the difference in intensity between the current and previous frames.

The combination of feature tracking and feature matching algorithms gets over the limitations of matching the features on the dominant plane, the rejection of outliers in the feature matching process can be represented mathematically using the random sample consensus (RANSAC) algorithm. Given a set of observed data points that contain outliers, RANSAC estimates the parameters of a mathematical model that best fits the inliers (the non-outlier data points). The algorithm starts by selecting a random subset of the data points and using them to fit the model. Then, it checks the remaining data points to see if they are consistent with the model. If a sufficient number of data points are found to be consistent with the model, the algorithm refits the model using all of the inliers. This process is repeated multiple times to find the best fit model that has the most inliers. The dominant plane matches from the previous frame can be used to guide the search for inliers in the current frame, reducing the computation time and increasing the accuracy of the feature matching process.

A 16x16 grid is generated using the bundled paths algorithm, which wraps each frame with the previous frame, by generating a camera path for each cell in the grid. The bundled paths method reduces the perspective distortions, and it deals with parallax.

Bundled-path stabilization has two main advantages: it can manage parallax and correct significant perspective distortions that occur when a single homography is used.

In the final step of the framework, the grid-based detection method is used to detect features in the video. The mathematical representation of this can be expressed as:

$$f = g(I, T)$$

Where I is the current frame, T is the threshold value for the grid, and g is a function that calculates the features in the video based on the intensity values in the current frame and the threshold value for the grid. The function g can be expressed as:

$$g\ (I, T) = \{p1, p2, \ldots, pn\}$$

Where p1, p2, ..., pn are the feature points detected in the current frame based on the intensity values and the threshold value T.

4.3 Object Detection Phase

YOLOv4 (You Only Look Once version 4) is a state-of-the-art object detection algorithm that has several advantages over other object detection models. One major advantage of YOLOv4 is its speed. YOLOv4 can process images in real-time, meaning it can detect objects in a video stream at a speed of around 40 frames per second on a GPU. This makes it ideal for applications that require fast and accurate object detection, such as surveillance, traffic monitoring, and robotics.

Another advantage of YOLOv4 is its accuracy. YOLOv4 has been shown to outperform other object detection models on several benchmark datasets, achieving state-of-the-art performance in terms of both accuracy and speed. This is due in part to the use of a highly optimized CNN architecture with many residual blocks, which allows YOLOv4 to extract features from the input image more efficiently than other models.

In addition, YOLOv4 is highly configurable and can be customized to meet the specific needs of different applications. For example, users can adjust the size of the input image, the number of cells in the grid, the number of bounding boxes per cell, and the threshold for objectness scores to optimize the model for different use cases.

Overall, YOLOv4 is a highly effective and efficient object detection algorithm that is well-suited for a wide range of applications. Its combination of speed, accuracy, and configurability

make it an attractive choice for researchers and practitioners who need fast and reliable object detection.

YOLOv4 divides the input image into a grid of cells and predicts a fixed number of bounding boxes and associated objectness scores for each cell. The model also associates each bounding box with class probabilities for each class of object it has been trained on.

To extract features from the input image, YOLOv4 uses a convolutional neural network (CNN) architecture with many residual blocks. The CNN features are used to predict the objectness scores, bounding box coordinates, and class probabilities for each cell and bounding box.

For objectness score prediction, YOLOv4 uses logistic regression, represented by the equation: P(object) = sigmoid(score), where "score" is the output of the final layer of the CNN for that bounding box.

For bounding box coordinate prediction, YOLOv4 predicts the coordinates relative to the coordinates of the cell in which the box is located, using the following equations: bx = sigmoid(tx) + cx, by = sigmoid(ty) + cy, bw = pw * exp(tw), bh = ph * exp(th). Here, "tx" and "ty" are the predicted x and y offsets of the center of the bounding box relative to the cell, "tw" and "th" are the predicted widths and heights of the box, "pw" and "ph" are the width and height of the anchor box (used to normalize the width and height predictions), and "cx" and "cy" are the coordinates of the top-left corner of the cell.

In addition to that, for class probability prediction, YOLOv4 uses softmax regression, represented by the equation:

$$P(class_i|object) = exp(score_i)/sum(exp(score_j))$$

where "score_i" is the output of the final layer of the CNN for class i, and "sum(exp(score_j))" is the sum of the exponential scores for all classes.

4.4 Webots simulation environment

This chapter presents a proposed fog-based system for real-time video stitching using four DJI Mavic Pro drones in the Webots robot simulator. The system harnesses the capabilities of fog computing to enable efficient local processing and analysis of video data at the edge. The research focuses on the implementation and optimization of the video stitching algorithm, which is seamlessly integrated into the fog-based system. Additionally, the Robot Operating System (ROS) is deployed on a local server, serving as the central hub for the fog-based system.

The proposed fog-based system consists of four DJI Mavic Pro drones, each equipped with a Python code and the DJI Software Development Kit (SDK) to capture and stream video. The fog computing paradigm is employed to enable seamless communication between the drones and the local server, which acts as a fog node. This communication is facilitated through the use of the ROS communication protocol, ensuring efficient and low-latency data transfer.

The local server, as a fog node, receives the video streams from the drones and applies the video stitching algorithm in real-time. To optimize the processing time and ensure a balanced workload distribution, a load balancing technique is implemented. This technique efficiently distributes the processing tasks among the fog nodes, maximizing the system's performance.

The proposed fog-based system offers several notable advantages over traditional centralized processing methods. Firstly, it reduces the reliance on cloud services, enabling real-time data processing and analysis at the edge. This is particularly advantageous for applications that involve video streaming and analysis, where immediate insights are crucial.

Secondly, the system ensures low-latency and efficient data transfer, which is essential for real-time applications. By leveraging fog computing, the video streams from the drones are processed locally, eliminating the need to transmit large amounts of data to a centralized cloud server. This results in reduced latency and improved overall system performance.

Lastly, the fog-based system offers scalability and flexibility. Additional fog nodes or drones can be easily added to the system, allowing for seamless expansion and adaptation to various application scenarios. This scalability ensures that the system can accommodate the growing demands of video stitching and other related tasks.

The proposed fog-based system utilizing four DJI Mavic Pro drones in the Webots simulator provides an efficient and scalable platform for real-time video stitching. It highlights the potential of fog computing in edge environments, where localized processing and analysis of video data can significantly enhance performance and reduce dependencies on cloud services.

Future work can focus on further optimization of the video stitching algorithm to improve the system's efficiency and accuracy. Additionally, the integration of additional sensors and testing the system in real-world scenarios would provide valuable insights into its practical applicability. Furthermore, the simulation tool videos generated in this research can serve as a valuable dataset for video stitching, utilizing the power of fog computing to process and enhance the stitching outputs.

Figure 4-7 Illustrates the simulation with four drones.

4.5 Summary

This chapter presented a detailed explanation of each phase of the proposed model. The proposed model aims to enhance the quality of live streams from multiple quadcopter cameras by stitching and stabilizing them. To achieve this, the model estimates the inter and intra motion between the camera feeds and frames, respectively. The optimization problem is used to determine the best fit stitching and stabilization of the video. The intra motion approach ensures temporal smoothness between frames of the same video, while the inter motion method ensures spatial alignment between multiple videos captured by the drones. To handle parallax scenes, each video frame is divided into smaller cells, making it easier to use the bundled-path methodology. The following chapter will present the experimentation, the results along with the discussion of the results.

CHAPTER FIVE
EXPERIMENTATION AND RESULTS

5 EXPERIMENTATION AND RESULTS

This chapter presents the results of the experiments that have been conducted to evaluate the proposed stitching model and also the evaluation of the object detection model. The following subsections will explain in detail the available and used datasets, the evaluation metrics as well as experimental setup and results.

5.1 Dataset

Datasets can be of various sizes, ranging from small datasets that can be managed on a single computer to massive datasets that require distributed computing systems to process. object detection or image classification tasks, a ground truth dataset may be created by manually annotating images with labels that specify the location and category of objects within the image. This ground truth dataset can then be used to train and evaluate machine learning models.

5.1.1 Video Stitching Dataset

The model used videos from which were used in other papers to compare the results of the stitching quality with other paper [19], the dataset used includes six videos which represents videos captured while moving which introduces shaky frames and jitters due to the temporal resolution.

In the case of capturing input videos using several moving cameras, such as cellphones or UAVs, the resulting video streams may contain a significant amount of camera motion and changes in viewpoint. This can pose several challenges for real-time video stitching, as the stitching algorithm needs to accurately estimate the camera motion and align the different video streams in real-time.

The use of multiple moving cameras can provide a rich source of video data for real-time video stitching applications, but also introduces several challenges related to camera motion estimation and alignment. By using appropriate sensors and computer vision techniques, these challenges can be addressed, and accurate real-time video stitching can be achieved.

5.1.2 Object Detection Dataset

The Common Objects in Context (COCO) dataset is a widely used large-scale image recognition dataset that contains labeled images of various objects, scenes, and activities. The

dataset was introduced in 2014 by Microsoft Research and is maintained by the COCO Consortium, which includes academic and industrial organizations.

The COCO dataset contains over 330,000 images with more than 2.5 million object instances labeled with 80 different object categories, making it one of the largest and most diverse datasets available for object recognition tasks. The images in the dataset were collected from various sources, including Flickr and Microsoft Bing, and cover a wide range of scenes and activities, including indoor and outdoor scenes, people, animals, vehicles, and more.

Each image in the COCO dataset is annotated with object bounding boxes and object category labels, providing rich and detailed information about the objects in the images. The bounding boxes specify the location of the object within the image, while the category labels indicate the type of object (e.g., person, car, dog, etc.). The dataset also includes captions for each image, providing a description of the scene and the objects within it.

In addition to the object recognition task, the COCO dataset has been used for various other computer vision tasks, including image captioning, visual question answering, and image segmentation. The dataset is often used as a benchmark for evaluating the performance of computer vision models, and several popular deep learning models have been trained on the COCO dataset.

5.2 Evaluation Metrics

An evaluation metric is a standard measurement used to assess the performance or effectiveness of a particular system, algorithm, or model. In other words, it is a quantitative measure used to determine how well a system or model is performing in a particular task or application. The choice of evaluation metric depends on the specific task and the goals of the system or model being evaluated.

5.2.1 Runtime

The delay of the output video, we use Python's time module to record the time before and after the processing is done, and then compute the difference between these times.

5.2.2 Stability Score

The stability score which is a measure of the smoothness of a stitched video [47]. Here's an overview of the steps involved:

1. Track features on the stitched video: This can be done using feature detection and tracking algorithms such as Lucas-Kanade or Shi-Tomasi. The goal is to track features that are present in multiple frames of the stitched video.
2. Retain tracks with a length of greater than twenty frames: This is done to filter out short-lived tracks that may not be reliable.
3. Calculate the energy percentage of the lowest frequencies (2nd to 6th without DC component) for each track: This can be done using techniques such as Fourier transforms or wavelet transforms. The goal is to quantify the smoothness of the track by measuring the energy content at different frequency bands.
4. Average the energy percentages from all tracks to obtain the final stability score: This provides an overall measure of the smoothness and stability of the stitched video. A high stability score (close to 1) indicates a smooth and stable video, while a low stability score indicates a shaky or unstable video.

A high stability score (close to 1) indicates a smooth and stable stitched video.

$$S_{\text{stability}} = 1 - \frac{\sum_{i \atop f_i > 0.1} |\text{FFT}(f)_i|}{\sum |\text{FFT}(f)_i|}$$

5.2.3 Stitching Score

The stitching score which is a measure of the quality of the stitching in a stitched video [48]. Here's an overview of the steps involved:

1. Calculate the feature reprojection error for each frame: This involves detecting and matching features in adjacent frames, and then transforming the features in one frame to the coordinate system of the other frame using the estimated transformation parameters. The reprojection error is then calculated as the distance between the transformed feature and the corresponding feature in the other frame.
2. Calculate the stitching score for a single frame: This is done by averaging the reprojection errors for all feature pairs in the frame.
3. Obtain the final stitching score by taking the worst score among all frames: This provides an overall measure of the quality of the stitching across the entire video. The worst score is used to identify the frame with the most significant stitching errors, which can be useful for further analysis and improvement.

A low stitching score indicates a good alignment and high-quality stitching.

$$S_{\text{stitching}} = \frac{1}{N} \sum_{i=1}^{N} e_i$$

5.2.4 Mean Average Precision

Mean Average Precision (mAP) is a commonly used evaluation metric in the field of computer vision and object detection. It measures the accuracy of an object detection model by comparing its predicted bounding boxes with the ground truth bounding boxes.

Here's an overview of the steps involved in calculating mAP:

1. For each class of objects, calculate the precision-recall curve: This involves sorting the predicted bounding boxes for that class based on their confidence scores, and then computing the precision and recall for each threshold level. Precision is the ratio of true positives to the total number of predicted positives, while recall is the ratio of true positives to the total number of ground truth positives.
2. Compute the Average Precision (AP) for each class: This is done by calculating the area under the precision-recall curve for that class.
3. Compute the mAP across all classes: This is done by taking the average of the AP values for all classes.

The mAP is a value between 0 and 1, where a higher value indicates better performance. A perfect model would have a mAP of 1, indicating that all predicted bounding boxes perfectly match the ground truth bounding boxes.

$$mAP = \frac{1}{N} * \sum_{i=1}^{N} \text{APi}$$

mAP is a useful metric for comparing the performance of different object detection models, and for identifying areas for improvement in a given model. However, it's important to keep in mind that mAP is just one of many possible evaluation metrics, and that its relevance and applicability may depend on the specific use case and requirements.

5.2.5 Accuracy

Accuracy is a commonly used evaluation metric in machine learning, and it measures how often a classifier correctly predicts the class label of a given instance. It's a simple and intuitive metric that can provide a quick assessment of model performance, but it has some limitations.

The accuracy is calculated by dividing the number of true positives and true negatives by the total number of objects. True positives are instances where the model correctly predicted a positive class label, while true negatives are instances where the model correctly predicted a negative class label. False positives are instances where the model predicted a positive class label, but the actual class label was negative, while false negatives are instances where the model predicted a negative class label, but the actual class label was positive, see figure 5-1 that shows the actual and predicted classes.

1. True Negative (TN) is when the model makes a correct classification by predicting the negative class values (real) to be negative (real).
2. False Positive (FP) is when the model makes a false classification by predicting the negative values (real) to be positive (fake).
3. False Negative (FN) is when the model makes a false classification by predicting the positive values (fake) to be negative (real).
4. True Positive (TP) is when the model makes a correct classification by predicting the positive class values (fake) to be positive (fake).

$$\text{Accuracy} = \frac{TP + TN}{TN + FN + FP + TP}$$

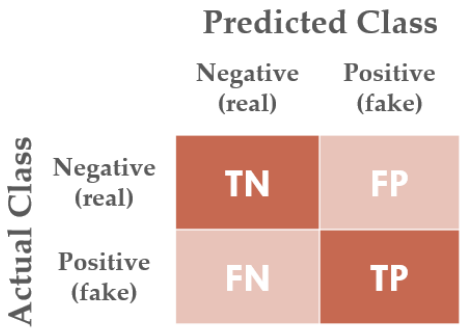

Figure 5-1 Shows the actual classes and predicted classes

5.3 Experimental Setup

The Raspberry Pi 4 serves as a powerful experimental environment, providing a quad-core 64-bit ARM Cortex-A72 CPU running at 1.5GHz and up to 8GB of RAM for testing and developing various computing applications. With its dual micro-HDMI ports supporting up to 4K resolution, Gigabit Ethernet, dual-band 802.11ac wireless, Bluetooth 5.0, and four USB ports, it offers a range of connectivity options for interfacing with other devices and peripherals. Its compatibility with multiple operating systems, including various Linux distributions and Windows 10 IoT Core, makes it an ideal platform for experimentation and prototyping. Whether for hobbyist projects, educational endeavors, or professional development, the Raspberry Pi 4 provides a flexible and accessible environment for exploring computing possibilities.

5.4 Experimental Work

In this thesis, the input videos were taken from the Tello drones, and the processing is done using RPI 4.

5.4.1 First Scenario Results: Static Object, Static Cameras

The scenario of static objects and static cameras is a common one, particularly in the field of image stitching. This scenario involves capturing multiple images of the same scene using cameras that are fixed in position and do not move during the image capture process. The objects in the scene are also static and do not move between the different images captured, see figure 5-2.

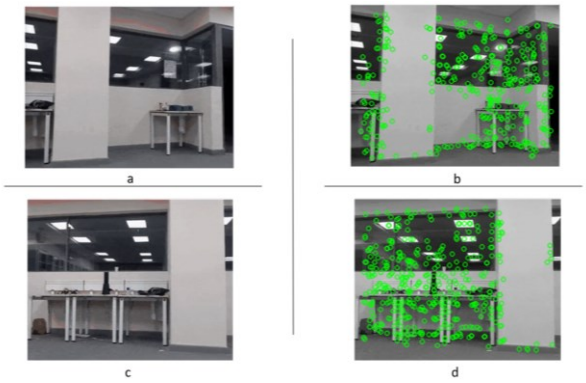

Figure 5-2 a. Illustrates the right view, b. illustrates the right view with feature detection, c. Illustrates the left view, d. illustrates the left view with feature detection.

This scenario is often encountered in applications such as panoramic photography, where multiple images of a scene are stitched together to create a single large image that captures a wider field of view than a single image. Other applications of this scenario include surveillance systems, where multiple cameras are used to cover a wide area.

The main advantage of this scenario is that it simplifies the image processing task, as the positions and orientations of the cameras are fixed and known, and the objects in the scene do not move. This allows for more accurate and efficient image stitching, as well as other types of image processing such as object detection, tracking, and recognition.

Overall, the scenario of static objects and static cameras is an important and widely used one in computer vision and is particularly useful in the context of image stitching and other image processing tasks, see figure 5-3.

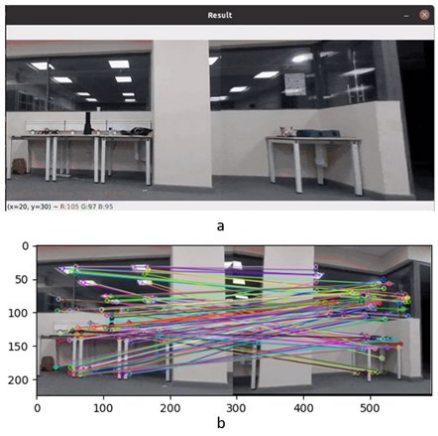

Figure 5-3 a. Illustrates the panorama view; b. illustrates the panorama view with feature matching.

5.4.2 Second Scenario Results: Moving Object, Static Cameras

In the scenario of moving objects and static cameras involves capturing images of a scene using cameras that are fixed in position and do not move during the image capture process, but where the objects in the scene are in motion, see figure 5-4 and figure 5-5.

This scenario is encountered in a wide range of applications, including security and surveillance systems in large buildings or outdoor areas, where multiple static cameras are positioned to cover the entire area. The resulting video provides a comprehensive view of the scene and helps in identifying potential security threats.

Another application of this scenario is event videography, where multiple static cameras are used to capture different angles of a performance, ceremony, or sports event. The resulting video provides a wide-angle view of the event, allowing the viewer to see all the important details.

The main challenge in this scenario is to accurately detect and track the moving objects in the scene, while also accounting for any changes in lighting, shadows, or occlusions. This requires the use of advanced computer vision techniques such as object detection, tracking, and recognition, as well as the ability to handle multiple object trajectories and occlusions.

Overall, the scenario of moving objects and static cameras is an important and challenging one in computer vision and is particularly useful in the context of security and surveillance systems, as well as event videography and other applications where a wide-angle view of a moving scene is required.

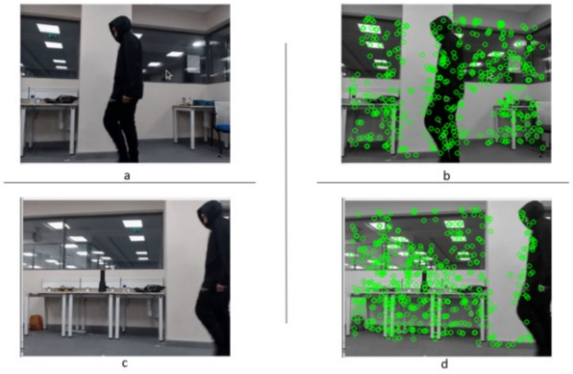

Figure 5-4 a. Illustrates the right view, b. illustrates the right view with feature detection, c. Illustrates the left view, d. illustrates the left view with feature detection.

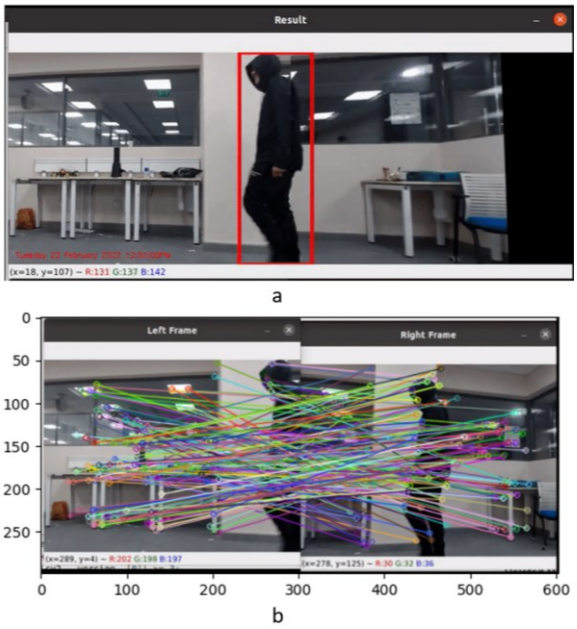

Figure 5-5 a. Illustrates the panorama view; b. illustrates the panorama view with feature matching.

5.4.3 Third Scenario Results: Static Object, Moving Cameras

The scenario of static objects and moving cameras involves capturing images of a scene using cameras that are in motion, but where the objects in the scene are static and do not move during the image capture process, see figure 5-6 and figure 5-7.

This scenario is encountered in a variety of applications, including virtual tours of real estate properties, where a camera mounted on a drone, or a wearable device is used to capture video as the operator moves through the property. The resulting video provides a seamless view of the property, giving the viewer an immersive experience.

Documentary filmmaking is another application of this scenario, where a moving camera is used to capture the entire story. The resulting video provides a seamless view of the scene and helps in creating an immersive experience for the viewer. Action sports is another popular application of this scenario, where a moving camera is used to capture the entire performance, allowing the viewer to see all the important details from multiple angles.

The main challenge in this scenario is to achieve video stabilization, which is the process of removing camera shake and improving the quality of the final video. This can be achieved through the use of advanced computer vision algorithms that analyze the motion of the camera

and compensate for any movements or vibrations, resulting in a smoother and more immersive viewing experience.

Overall, the scenario of static objects and moving cameras is an important and challenging one in computer vision and is particularly useful in the context of virtual tours, documentary filmmaking, and action sports. It requires advanced techniques for video stabilization and motion analysis and can provide a more immersive viewing experience for the viewer.

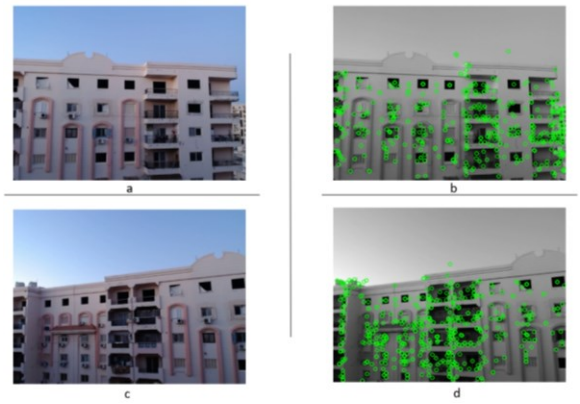

Figure 5-6 a. Illustrates the right view, b. illustrates the right view with feature detection, c. Illustrates the left view, d. illustrates the left view with feature detection.

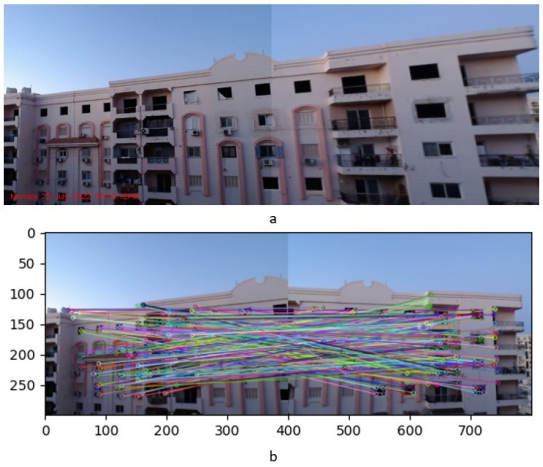

Figure 5-7 a. Illustrates the panorama view; b. illustrates the panorama view with feature matching.

5.4.4 Experimental Results

Table 2, based on the results of the three trials, we can see that the stability score was consistently high, with the first trial having a perfect score of 1.00 and the other two trials having

scores of 0.93 and 0.90, respectively. This indicates that the stitched videos were relatively smooth and stable across all three experiments.

Table 5-1 The comparison of the "Stability Score" and "Stitching Score" for different experimental scenarios

	Scenario I	Scenario II	Scenario III
Stability Score	1.00	0.93	0.90
Stitching Score	0.67	1.01	1.02

On the other hand, the stitching score varied more across the three trials, with the first trial having a score of 0.67, which is a relatively low score indicating good alignment and high-quality stitching. The second and third trials had higher stitching scores of 1.01 and 1.02, respectively, indicating lower quality stitching with lower alignment than the first trial.

Given that the three experiments were independent and conducted in different environments, it is interesting to note that the stability score remained high across all three trials. However, the stitching score varied, which suggests that the quality of the stitching may be influenced more by the specific environmental factors and conditions of each experiment, rather than the overall stability of the stitched video.

Overall, a high stability score is desirable as it indicates a smoother and more stable stitched video, while a lower stitching score is also desirable as it indicates better alignment and higher quality stitching.

5.4.5 Real Time Video Stitiching Comparison Results

The stability and stitching scores for the examples tested are summarized in Table 3, based on these scores, it appears that there has been a significant improvement for all of the examples.

Table 5-2 The comparison of the "Stability Score" and "Stitching Score" for different experimental scenarios

Video	Length	Frame width	Frame height	Approx. fps	Stitching score		Stability score		
Video 1	0:00:13	1280	720	30 fps	0.99	0.9	0.67	0.83	0.88
Video 2	0:00:12	1280	720	30 fps	0.54	0.51	0.37	0.78	0.82
Video 3	0:00:10	1280	720	30 fps	1.07	1.01	0.65	0.82	0.86
Video 4	0:00:13	1280	720	30 fps	1.01	0.89	0.71	0.88	0.89
Video 5	0:00:15	1280	720	30 fps	0.43	0.37	0.68	0.84	0.88
Video 6	0:00:12	1280	720	30 fps	1.04	0.94	0.63	0.91	0.94

5.4.5 Object Detection Results

If an object detection model has a mAP value of 89.5%, it means that, on average, the precision at different recall levels is 89.5%.

If an object detection model has an accuracy of 93.28%, it means that, on average, it correctly identified and localized 93.28% of the objects in the images it was tested on.

Prediction time is a measure of how long it takes for the model to process an image and output its predictions. In the case of the object detection model, it takes 4.9 milliseconds on average to process an image and output its predictions.

So, in summary, an object detection model with a mAP value of 89.5%, an accuracy of 93.28%, and a prediction time of 4.9 milliseconds is a high-performing and efficient model that can identify and localize objects in images with high precision and speed, see figure 5-8.

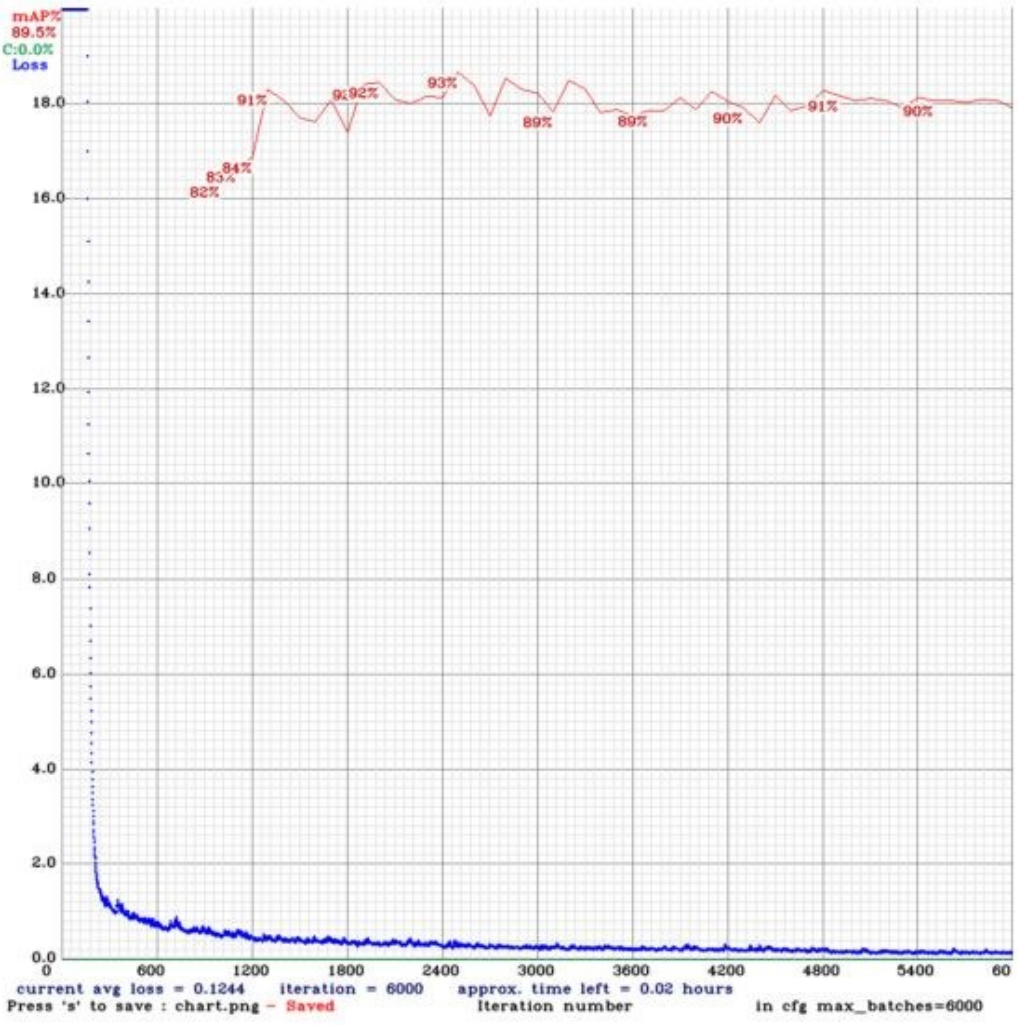

Figure 5-8 mAP and loss graph for object detection model

5.5 Summary

This chapter presents the results of the proposed model. Different performance measures were reported such time delay, stitching score and stability score on the stitching level and mAP and accuracy on the object detection level. The suggested model displayed exceptional object detection performance, achieving high average precision measures across multiple recall levels. Moreover, it exhibited consistent and robust performance in diverse experimental settings, with relatively high stability and stitching scores which indicates how the model is effective. A high stability score is desirable as it indicates a smoother and more stable stitched video, while a lower stitching score is also desirable as it indicates better alignment and higher quality stitching. An object detection model with a mAP value of 89.5%, an accuracy of 93.28%, and a prediction time of 4.9 milliseconds is a high-performing and efficient model that can identify and localize objects in images with high precision and speed.

CHAPTER SIX

CONCLUSION AND FUTURE WORK

6 CONCLUSION AND FUTURE WORK

This chapter briefly summarizes and discusses the work completed, followed by conclusions. The chapter ends up with a discussion of some directions for the future work.

6.1 Conclusion

With the advancement of technology and the emergence of new requirements, image and video stitching has become an essential aspect of both personal and professional applications. To address the challenges and opportunities presented by this field, various experiments have been conducted. The outcomes demonstrate that the proposed approach is capable of producing panoramic images with improved compression ratios, faster and more precise reconstruction, and enhanced object detection capabilities.

The proposed model works on jointly stitching and stabilizing the live stream from two or more quadcopters. The estimation of inter motion between the live feeds from the cameras and intra motion between the frames of the same video. The entire process is turned into an optimization problem to get the best fit stitching and stabilized video, so the intra motion method assures the temporal smoothness sustainability between the different frames of the same video, and the inter motion method assures the forcing of the spatial alignment between the multiple videos provided by the drones. Handle scenes with parallax, each video frame is divided into smaller cells so that it is easier to use the bundled-path methodology.

The suggested model displayed exhibited consistent and robust performance in diverse experimental settings, with relatively high stability and stitching scores which indicates how the model is effective.

In all three trials, the stability score remained consistently high, with the first trial achieving a perfect score of 1.00 and the other two trials achieving scores of 0.93 and 0.90, respectively. This indicates that the stitched videos were smooth and stable across diverse experimental settings. However, the stitching score varied more across the three trials, with the first trial having a score of 0.67, indicating high-quality stitching with good alignment. On the other hand, the second and third trials had higher stitching scores of 1.01 and 1.02, respectively, indicating lower quality stitching with lower alignment than the first trial. The variation in stitching scores

suggests that the quality of stitching is influenced more by specific environmental factors and conditions of each experiment than the overall stability of the stitched video. In conclusion, a high stability score signifies a smoother and more stable stitched video, while a lower stitching score indicates better alignment and higher quality stitching.

In addition, the proposed object detection model showed outstanding performance, achieving high average precision measures across various recall levels. Specifically, the model exhibited a mAP value of 89.5%, an accuracy of 93.28%, and a prediction time of 4.9 milliseconds, making it a highly efficient and accurate model capable of accurately identifying and localizing objects in images with great precision and speed.

6.2 Future Work

The current study has demonstrated the effectiveness of a proposed model for real-time video stitching and object detection using quadcopter cameras. While the results are promising, there are still opportunities for future work in this field.

Future work in this area could focus on developing algorithms and techniques that can effectively handle the challenges presented by moving cameras and moving objects. This could involve exploring different approaches to video stabilization and motion estimation, as well as developing more sophisticated object detection models that can accurately detect and track moving objects in real time.

One other potential avenue for further exploration is the use of more advanced object detection techniques, such as deep learning-based models. These models have shown great promise in recent years and may be able to improve upon the already impressive performance demonstrated by the current model.

Overall, the current study provides a strong foundation for future work in this field and demonstrates the potential for drones to be used in a variety of applications requiring real-time video stitching and object detection.

REFRENCES

REFERENCES

[1] P. A. D. P. S. S. K. V. Chung A, "A survey on aerial swarm robotics," IEEE Trans Robot., 2018.

[2] T. J. L. S. Xi, "Review of unmanned aerial vehicle swarm communication architectures and routing protocols," Appl Sci, 2020.

[3] A. Tahir, J. Böling and M.-H. e. a. Haghbayan, "Swarms of unmanned aerial vehicles — A survey. In: Journal of Industrial Information Integration.," 2019.

[4] M. a. B. M. Dorigo, "Swarm intelligence," 2007.

[5] Z. Z. L. C. Y. Z. Wei LYU, "A survey on image and video stitching," 2019.

[6] S. Y. a. S. F. M. el Shehaby, ""An Efficient Multi-View Panoramic Imaging and Extra Compression of Surveillance Cameras' Footage Using Stitching,"," 2019.

[7] A. &. R. R. &. N. P. &. M. M. &. C. V. &. Y. F. &. G. M. Jain, "AI-Enabled Object Detection in UAVs: Challenges, Design Choices, and Research Directions," 2021.

[8] K. &. L. S. &. C. L.-F. &. Z. B. Lin, "Seamless Video Stitching from Hand-held Camera Inputs," 2016.

[9] J. J. &. B. A. Flores, "Let's Democratize Drones! Using the Ryze Tello Drone as a Tool for Ecological Farm Design & Landscape Ecology Research.," 2019.

[10] J. A. K. V. P. G. Ahmadzadeh A, "Multi-UAV cooperative surveillance with spatio-temporal specifications.," Proceedings of the 45th IEEE conference on decision and control, 2006.

[11] V. V. S. M. Petrl´ık M, "Coverage optimization in the cooperative surveillance task using multiple micro aerial vehicles," IEEE international conference on systems man and cybernetics (SMC), 2019.

[12] T. M. K. A. M. D. M. N. K. V. Mohta K, "QuadCloud: a rapid response force with quadrotor teams.," Cham: Springer International Publishing, 2016.

[13] V. V. C. J. T. J. L. G. Saska M, "Swarm distribution and deployment for cooperative surveillance by micro-aerial vehicles.," J Intell Robot Syst., 2016.

[14] D. R. Ritz R, "Carrying a flexible payload with multiple flying vehicles," IEEE/RSJ international conference on intelligent robots and systems, 2013.

[15] K. V. Loianno G, "Cooperative transportation using small quadrotors using monocular vision and inertial sensing.," IEEE Robot Autom Lett. , 2018.

[16] S. M. G. M. C. N. C. V. C. C. Abdelkader M, "Optimal multi-agent path planning for fast inverse modeling in UAV-based flood sensing applications.," international conference on unmanned aircraft systems (ICUAS), 2014.

[17] J. W. a. S.-F. C. Y. Wang, ""Camswarm: Instantaneous smartphone camera arrays for collaborative photography."," 2015.

[18] W. J. a. J. Gu, ""Video stitching with spatial-temporal content preserving warping," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)," 2015.

[19] S. L. T. H. S. Z. B. Z. G. M. Heng Guo, "Joint Video Stitching and Stabilization From Moving Cameras. IEEE Trans Image Process.," 2016.

[20] W. X. J. Z. M. Z. Z. W. a. X. L. J. Li, ""Efficient Video," 2015.

[21] F. Perazzi et al., ""Panoramic video from unstructured camera arrays,"," 2015.

[22] H. L. P. T. G. Z. a. H. B. H. Jiang, ""3D reconstruction," 2012.

[23] M. B. W. T. L. A. P. a. J.-P. G. Martin Andreoni Lopez, "Towards Secure Wireless Mesh Networks for UAV Swarm Connectivity: Current Threats, Research, and Opportunities," 2021.

[24] J. Y. a. D. Lee, "Real-Time Video Stitching Using Camera Path Estimation and Homography Refinement," 2017.

[25] T. &. J. F. &. L. J. Yang, "A fast and robust real-time surveillance video stitching method," 2020.

[26] P. &. S. D. &. B. S. Shete, "Real-time panorama composition for video surveillance using GPU," 2016.

[27] B.-S. &. C. K.-A. &. P. W.-J. &. K. S.-W. &. K. S.-J. Kim, "Content-preserving video stitching method for multi-camera systems," 2017.

[28] K.-B. &. Z. Y. Jia, " Multi-camera video stitching based on foreground extraction," 2012.

[29] S. &. C. U. &. K. P. &. B. H. &. A. A. &. V. D. Degadwala, "Real-Time Panorama and Image Stitching with Surf-Sift Features," 2021.

[30] Z. &. L. Y. &. C. X. &. Y. T. &. W. W. &. W. M. &. D. R. Bai, "Real-Time Video Stitching for Mine Surveillance Using a Hybrid Image Registration Method," 2020.

[31] M. &. Y. S. &. F. S. El Shehaby, "An Efficient Multi-View Panoramic Imaging and Extra Compression of Surveillance Cameras' Footage Using Stitching," 2)19.

[32] L. &. Z. Z. &. L. C. &. Z. Y. Wei, "A survey on image and video stitching," 2019.

[33] H. &. L. S. &. H. T. &. Z. S. &. Z. B. &. G. M. Guo, "Joint Video Stitching and Stabilization From Moving Cameras," 2016.

[34] R. &. W. Z. &. X. Z. &. W. C. &. L. Q. &. Z. Y. Wang, "A Real-time Object Detector for Autonomous Vehicles Based on YOLOv4.," 2021.

[35] Z. &. Z. L. &. L. S. &. J. Y. Jiang, "Real-time object detection method based on improved YOLOv4-tiny.," 2020.

[36] A. &. W. C.-Y. &. L. H.-y. Bochkovskiy, "YOLOv4: Optimal Speed and Accuracy of Object Detection.," 2020.

[37] DJI, "github," 15 Feb 2019. [Online]. Available: https://github.com/dji-sdk/Tello-Python.

[38] "raspberrypi," [Online]. Available: https://www.raspberrypi.com/documentation/.

[39] "docs.opencv," [Online]. Available: https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_fast/py_fast.html.

[40] "docs.opencv," [Online]. Available: https://docs.opencv.org/master/df/d0c/tutorial_py_fast.html.

[41] "docs.opencv," [Online]. Available: https://docs.opencv.org/3.4/d4/dee/tutorial_optical_flow.html.

[42] R. I. &. Z. A. Hartley, "Multiple view geometry in computer vision.," *Cambridge University Press,* 2004.

[43] "baeldung," [Online]. Available: https://www.baeldung.com/cs/ransac.

[44] "roboflow," [Online]. Available: https://blog.roboflow.com/a-thorough-breakdown-of-yolov4/.

[45] Z. L. L. v. d. M. Gao Huang, "Densely Connected Convolutional Networks," 2018.

[46] M. T. R. P. Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection," 2020.

[47] L. Y. P. T. a. J. S. S. Liu, "Bundled camera paths for video stabilization," *ACM Trans,* vol. 32, p. 78, 2013.

[48] M. B. a. L. M. C. Buehler, "Non-metric image-based rendering for video stabilization," *IEEE CVPR,* vol. 2, 2001.

[49] M. S. W. a. L. Andrew, "Drone swarms—A monograph by school of advanced military studies.," 2017.

[50] M. B. a. D. G. Lowe, "Automatic Panoramic Image Stitching using Invariant Features," 2007.

[51] S. &. J. J. &. K. D. &. K. Y. Moon, "Swarm Reconnaissance Drone System for Real-Time Object Detection Over a Large Area.," 2023.

APPENDICES

APPENDIX A: LIST OF PUBLICATIONS DERIVED FROM THE THESIS

"Natural-Inspired Drone Swarm Processing Fov for Efficient Multi-View Monitoring and Object Detection", International Journal of Novel Research and Development, ISSN:2456-4184, Vol.8, Issue 3, page no. e157-e173, March-2023.

"Swarm Unmanned aerial vehicles (UAVs)-based Fog Computing Platform Supporting Internet of Things Applications", International Society for Photogrammetry and Remote Sensing, Egypt GSW'2023.

"A New Hybrid Model for Computer-Vision Video Stitching Using Intelligent Swarm Drones", the 8th International conference on Advanced Technology and Applied Sciences (ICaTAS'2023).

إقرار الباحث

أقر بأن المادة العلمية الواردة في هذه الرسالة قد تم تحديد مصدرها العلمي وأن محتوى الرسالة غير مقدم للحصول على أي درجة علمية أخرى، وأن مضمون هذه الرسالة يعكس أراء الباحث الخاصة وهي ليست بالضرورة الآراء التي تتبناها الجهـة المانحة.

الاسم: أسامة هشام السيد عبد الرازق

التوقيع: 

التاريخ: ٢٠-يوليو- ٢٠٢٣

خاتمة الرسالة

مع تقدم التكنولوجيا وظهور متطلبات جديدة، أصبحت عملية دمج الصور والفيديو جانباً أساسياً من التطبيقات الشخصية والمهنية. ولمعالجة التحديات والفرص التي يوفرها هذا المجال، تم إجراء مختلف التجارب. تشير النتائج إلى أن النهج المقترح قادر على إنتاج صور بانورامية بمعدلات ضغط محسّنة، وإعادة إنشاء أسرع وأدق، وقدرات تحسين الكشف عن الكائنات.

يعمل النموذج المقترح على دمج وتثبيت بث مباشر من طائرتين متعددة أو أكثر. يتم تقدير الحركة بين التغذية المباشرة من الكاميرات والحركة داخل إطارات نفس الفيديو. يتم تحويل العملية بأكملها إلى مشكلة تحسين للحصول على أفضل تناسب بين الدمج والفيديو المثبت. تضمن الطريقة الحركية الداخلية الاستمرارية الزمنية بين إطارات مختلفة من نفس الفيديو، في حين يضمن الحركة الخارجية الاتساق المكاني بين مقاطع الفيديو المتعددة المقدمة من الطائرات بدون انحراف.

لمعالجة المشاهد التي تحتوي على تحويم، يتم تقسيم إطار الفيديو إلى خلايا صغيرة حتى يكون من السهل استخدام منهجية المسار المجمع.

في الثلاث محاولات، ظلت درجة الاستقرار عالية بشكل متسق، حيث حققت المحاولة الأولى درجة استقرار مثالية تبلغ 1.00، في حين حققت المحاولتان الأخريان درجات استقرار تبلغ 0.93 و0.90 على التوالي. ويشير ذلك إلى أن الفيديوهات الملتصقة كانت سلسة ومستقرة في إعدادات تجريبية متنوعة. ومع ذلك، اختلفت درجة الخياطة أكثر بين الثلاث محاولات، حيث حققت المحاولة الأولى درجة خياطة تبلغ 0.67، مما يدل على خياطة عالية الجودة مع توافق جيد. وعلى الجانب الآخر، حصلت المحاولتان الثانية والثالثة على درجات خياطة أعلى تبلغ 1.01 و1.02 على التوالي، مما يشير إلى خياطة ذات جودة أقل وتوافق أقل من المحاولة الأولى. وتشير التباينات في درجات الخياطة إلى أن جودة الخياطة تتأثر أكثر بالعوامل البيئية وشروط كل تجربة بشكل أكبر من الاستقرار العام للفيديو الملتصق. وبشكل عام، تدل درجة الاستقرار العالية على فيديو ملتصق أكثر سلاسة واستقرارًا، في حين تشير درجة الخياطة المنخفضة إلى توافق أفضل وجودة خياطة أعلى.

بالإضافة إلى ذلك، أظهرت النموذج المقترح للكشف عن الأشياء أداءً متميزًا، حيث حقق قياسات دقة متوسطة عالية عبر مستويات التذكر المختلفة. على وجه التحديد، أظهر النموذج قيمة mAP بنسبة 89.5%، ودقة بنسبة 93.28%، ووقت توقع بمقدار 4.9 ملي ثانية، مما يجعله نموذجًا فعالًا ودقيقًا قادرًا على التعرف وتحديد مواقع الأشياء في الصور بدقة وسرعة عالية.

بناءً على ذلك، فإن الخيارات الحالية للتصوير البانورامي والتركيب الفيديو تعتمد على العديد من التقنيات المختلفة، بما في ذلك التصوير المتعدد الكاميرات، والتقاط الصور بزوايا مختلفة، والتراكب الذاتي، والتوافق الديناميكي. ومع ذلك، يبدو أن النموذج المقترح يوفر تحسينًا ملموسًا في الأداء الشامل لتركيب الصور والفيديو. يوفر النموذج المقترح تقنية تركيب متقدمة تعتمد على تحسين النسق الزمني وتحقيق توافق مكاني أفضل، بالإضافة إلى تحسين قدرة الكشف عن الكائنات في الصور والفيديو. وبالتالي، يمكن استخدام هذا النموذج في مجموعة واسعة من التطبيقات الشخصية والمهنية، بما في ذلك التصوير الجوي والفضائي وتسجيل الفعاليات الرياضية والحفلات الموسيقية وأكثر من ذلك.

ملخص الرسالة

في السنوات الأخيرة، زاد استخدام المركبات الجوية غير المأهولة أو الطائرات بدون طيار في مجالات مختلفة، بما في ذلك تطبيقات المراقبة. ومع ذلك، يمكن أن يكون استخدام عدة طائرات بدون طيار للمهام السرية مكلفًا ومعقدًا، مما يجعل من الصعب إدارة ومراقبة البيانات المولدة. في هذا الصدد، اقترح النموذج الذكي باستخدام طائرات السرب لتطبيقات المراقبة بديلاً فعالاً من حيث التكلفة باستخدام الطائرات التجارية المتاحة في الأسواق للتحكم المتزامن في عدة طائرات بدون طيار وخياطة الفيديو في الوقت الحقيقي. تهدف الدراسة إلى تقليل نقل البيانات والتخزين والإدارة والمراقبة من خلال تقديم تصوير متعدد المناظير فعال وضغط إضافي للفيديو الخاص بكاميرات طائرات السرب من خلال الخياطة.

لتحقيق هذا الهدف، استخدمت الدراسة طريقة تقدير مسار الكاميرا وتحسين الهوموغرافيا لتمكين التحكم المتزامن في عدة طائرات بدون طيار وخياطة الفيديو في الوقت الحقيقي. بالإضافة إلى ذلك، اقترح الباحثون إطار عمل موحد لخياطة واستقرار الفيديو المشترك، والذي يتضمن إنشاء مسار كاميرا افتراضي مثالي، وتحسين المدى الزمني للفضاء، وتتبع شبكة النقاط لتحسين الصلابة، ونماذج الحركة المستندة إلى الشبكة للتعامل مع التحولات الزاوية في مقاطع الفيديو التي يتم التقاطها. أجرت الدراسة ثلاثة تجارب لتقييم فعالية النهج المقترح. واستناداً إلى النتائج، كان مؤشر الاستقرار عالياً بشكل ثابت عبر جميع التجارب الثلاثة، مما يدل على خياطة فيديو سلسة ومستقرة نسبياً. ومع ذلك، تباين مؤشر الخياطة عبر التجارب الثلاثة، حيث كان للتجربة الأولى درجة خياطة منخفضة نسبياً، مما يدل على تحقيق مستوى جيد من التحديد وجودة عالية في الخياطة. بالمقابل، كانت للتجارب الثانية والثالثة درجات خياطة أعلى، مما يدل على جودة خياطة أقل مع تحديد أقل.

باختصار، تعد الطريقة المقترحة لاستخدام الطائرات بدون طيار المتوفرة تجارياً لتنفيذ مهام السرب مع تجميع الفيديو بشكل فوري وتثبيتها من الطرق الفعالة والاقتصادية لتطبيقات المراقبة. أظهرت نتائج الدراسة أن الطريقة حققت نتائج استقرار عالية وفي نفس الوقت حافظت على مستوى مرضي من جودة التجميع، مما يجعلها حلاً واعداً لإدارة ومراقبة الكميات الكبيرة من البيانات التي تولدها العديد من الطائرات

الأكاديمية العربية للعلوم والتكنولوجيا والنقل البحري

كليـــة الهندســـة والتكنولوجيـــــــا

قسم هندسة الحاسب

تقنية تجمعات الطائرات بدون طيار المستوحاة من الطبيعة للمراقبة المتعددة والكشف عن الكائنات بكفاءة

إعداد

أسامة هشام السيد عبد الرازق

بكالوريوس هندسة الحاسب 2019

كلية الهندسة والتكنولوجيا

الأكاديمية العربية للعلوم والتكنولوجيا والنقل البحري

رسالة مقدمة للأكاديمية العربية للعلوم والتكنولوجيا والنقل البحري لاستكمال متطلبات نيل درجة

ماجستير العلوم

في

هندسة الحاسب الآلي

تحت إشراف

أ.د. شيرين مصطفى يوسف **أ.د. أسامة اسماعيل**

قسم هندسة الحاسب

كلية الهندسة والتكنولوجيا

الأكاديمية العربية للعلوم والتكنولوجيا والنقل البحري

الإسكندرية

2023